PhD-FSTC-2014-41
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 18/11/2014 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Denis SHIRNIN

Born on 16 March 1983 in Bishkek (Russia)

FORMALISING THE TWOFOLD STRUCTURE OF A PROACTIVE SYSTEM: PROOF OF CONCEPT ON DETERMINISTIC AND PROBABILISTIC LEVELS

Dissertation defense committee

Dr. Denis Zampunieris, dissertation supervisor
Professor, Université du Luxembourg

Dr. Christoph Schommer, committee member
Associate Professor, Université du Luxembourg

Dr. Steffen Rothkugel, Chairman
Associate Professor, Université du Luxembourg

Dr. Jean-Noël Colin, committee member
Professor, Université de Namur, Belgium

Dr. Merik Meriste, Vice Chairman
Professor, University of Tallinn, Estonia

Abstract

In the present work, we investigate the domain of *Proactive computing*, which may be characterised as a comparatively new computer science research paradigm. In our study, we understand and use the notion of *Proactive Computing* as it was defined by Dr. David L. Tennenhouse [1, 2]. Conceptually, the architecture and main framework of our software system is entirely based on underlying concept characteristics of the given approach.

Being a new research paradigm, proactive computing doesn't possess yet a clear methodological support and approach repository, which may subsequently provide us with the necessary research techniques and methods. An application of the proactive computing approach thus requires a thorough empirical and theoretical investigation.

The main objective of the current research project is to design an effective methodological framework, which will subsequently allow us to provide our proactive system with a scientifically rationalised proof of concept. An acquisition of the scientific evidence is primarily accomplished through the design, implementation and testing of all composite theoretical, methodological and functional aspects of the proactive system's framework.

The present study aims to emphasise the importance of designing a coherent multidisciplinary methodological framework, which is initially based on the expertise of two research fields, computer science and cognitive science domains. The main composite aspects of our methodological structure are based on the specific multidisciplinary techniques from both research fields. We investigate the proactive computing paradigm within the defining principles of deterministic and probabilistic approach orientations, where we additionally apply the techniques of the cognitive modelling approach.

All designed and implemented computational mechanisms, together with the realised empirical studies, collectively aim to formalise the concept characteristics of the proactive system. Ultimately, an effective validation of the employed methodological frameworks and their underlying computational techniques provide our system with the necessary, scientifically rationalised proof of concept.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Professor Denis Zampunieris, whose expertise, continuous support and guidance allowed me to accomplish this work. I am grateful for his patience and encouragements, expressed throughout the years of the joint work.

I would like to thank the committee members Professor Christoph Schommer and Professor Merik Meriste for their advices and valuable contributions. I am thankful for their expertise, constructive recommendations and insightful comments, which have been a valuable asset in the realisation of the current work.

I express my thanks to Professors Steffen Rothkugel, Robert Reuter and Jean-Noël Colin for honouring me to be the members of the thesis committee. Furthermore, I would like to thank Professor Ulrich Sorger for the provided advices and assistance.

I express my special thanks to Sandro Reis and Sergio Marques Dias for their valuable technical support and collaborative research. I am grateful to all my colleagues, who in one way or another helped me or provided me with a valuable constructive criticism and friendly advice.

I would like to thank my family and friends for helping, encouraging and aspiring me throughout the years of my research endeavours.

Contents

List of Figures

List of Tables

Part I

Theory

Chapter 1

Introduction

The current chapter aims to provide the reader with the necessary primary information, related to the different aspects of the given thesis work. In the upcoming sections, we define the thesis' methodological directions, and we specify the study's main objectives and orientations. Together with a research problem definition, we consequently introduce the thesis' underlying research questions, which define the structure of the current work. Additionally, we find it necessary to acquaint the reader at the present introductory step with the key methodological characteristics. Thus, we outline below the governing aspects of the study's methodological and empirical specifications.

The present chapter is structurally divided into five composite parts. Collectively, all outlined introductory information aims to highlight the key orientations and the main governing points of the current thesis. Thus, in section 1.1, we delineate the purpose of the thesis, where we specify its background and underlying motivations. Consequently, in section 1.2, we introduce an associated study area and the thesis' main focus. In the following section 1.3, we specify the thesis' objectives and key research orientations. We acquaint a reader with the methodological and empirical characteristics of the study in section 1.4. Ultimately, we provide a detailed outline of the entire thesis structure in the corresponding section 1.5.

1.1 Purpose of the thesis

Below we present the related background information and prerequisite thesis' details, which conjointly define the main directions and orientations of the current research project. Moreover, the highlighted information serves to underline the importance of contributions and research efforts, made by previous and current scientific collaborators.

1.1.1 BLAST Project

Entitled as Better e-Learning Assignments System Technology, the BLAST[1] project has been initiated in 2006 by Professor Denis Zampunieris with the objective to build a prototype version of the proactive e-learning assistant, based on the currently employed University's Moodle platform [3–5]. The initial project objective was therefore to design and to implement the Proactive Online Assignments System for blended teachings at the University of Luxembourg for supporting students and teachers in their online academic activities. The given *Research & Development* project aimed to build a fully functional Online Assignments System with advanced proactive features, designated for both beginners and advanced users.

Thus, throughout the time and with the help of accorded research efforts, the project evolved into a broader area of interdisciplinary research, which currently encompasses the contributions of several scientific collaborators and junior researchers. Consequently, the Proactive Online Assignments System has evolved into a more conceptually refined and user-oriented Proactive System.

1.1.2 PhD thesis in the frame of the BLAST Project

The current PhD thesis is taking place in the frame of the BLAST project, and is therefore partially based on the previous research efforts, where it aims to provide an additional research input and to contribute to the interdisciplinary aspect of the ongoing *Research & Development* project. Bound together, the activities of the PhD thesis and the BLAST project represent an opportunity to produce constructive research results, issued from a cross-domain contribution of two different discipline fields: *Computer Science* and *Cognitive Science*.

The purpose of the current doctoral research is therefore to provide a necessary theoretical input from the cognitive science domain, which allows us to extend the methodological scope and proficiency level of the proactive system's conceptual framework. An effective formalisation of all theoretical and functional aspects of the system provides the current research with a consistent and viable proof of concept. An acquisition of the scientific evidence is primarily accomplished through the design, implementation and testing of all composite theoretical, methodological and functional aspects of the proactive system's framework. Consequently, an effective realisation of a scientifically rationalised proof of concept represents the study's governing objective and the thesis' main purpose.

[1] The "BLAST–Better e-Learning Assignments System Technology" project was supported by the University of Luxembourg under the grant no. F1R-CSC-PUL-10BLAS during the years 2011–2012.

1.2 Study area

Throughout the years of the present doctoral research, numerous theoretical concepts and empirical approaches have been studied, where some of them have been selected, analysed and subsequently implemented into the methodological framework of the current work. The theoretical key orientation is represented by the computer science research field of *proactive computing*. The given field defines the underlying aspects of the thesis' conceptual specifications and their corresponding methodological structure. All supplementary theoretical frameworks, together with their associated empirical techniques, ultimately aim to support the methodological structure and functional aspects of the proactive computing research.

The current work investigates the domain of *Proactive computing*, which may be initially defined as a comparatively new computer science research field [2]. Conceptually, the architecture and main framework of our software system is entirely based on the underlying concept characteristics of the given approach. A concept of *proactivity*, applied within the domain of computing technologies may be largely defined as an anticipation of a context event, activity-based problem, unwanted event, context conditions, or context change [6, 7]. A system, which employs the intrinsic principles of proactive computing aims to detect the aforementioned elements in order to be able to act proactively with regard to its initial objectives and the conditions of a currently prevailing context setting.

Consequently, we conduct our study and build its corresponding methodological framework on the basis of the aforementioned field of proactive computing research. More precisely, we understand and use the notion of *Proactive Computing* as it was initially defined by Dr. David L. Tennenhouse [1, 2]. In his vision of a future of the computer science research he explores a new paradigm, where he proposes to re-examine the relationships between physical and abstract domains of a context, to re-evaluate our general approach of interactive computing, and to gradually move from interactive human-centred to proactive human-supervised models of human-computer interactions. By taking into consideration the relationships between the associated aspects of physical and abstract domains of a context, and through the enhancement of a target system by a proactive type of behaviour, we are therefore able to build more intuitive and perceptive software systems.

Essentially, the key concept of proactive computing and its compound underlying principles characterise the proactive system as an anticipatory context monitoring mechanism, which aims to accurately assess a target situation and to successively perform the necessary context mediating actions. In other words, a system, built within a framework of the proactive computing paradigm primarily operates within the specified target environment, where it aims to anticipate or prevent certain context events by continuously monitoring, and if necessary mediating the composite aspects of a context setting [8, 9]. As we will see in our upcoming chapters, the context composite aspects are often characterised by hidden human factors, which pervade the space of context details.

Thus, a target context situation is often composed of numerous "undetectable" aspects of human behavioural characteristics and users' cognitive variations, which play an essential role in the accurate analysis of context settings. Consequently, in order to allow the system to be perceptive towards hidden aspects of the human behaviour, we have to build the appropriate mechanisms, which will be able to penetrate into the subtle abstract layers of the associated context details. Ultimately, in order to effectively realise the fulfilment of the given objective, we must employ in our research the knowledge and expertise of the associated cognitive science domain. Only then we will be able to endow our system with the required perceptive qualities, capable to detect the subtle aspects of a target context environment.

The present study aims to emphasise the importance of designing a coherent and adapted multidisciplinary methodological framework, which is initially based on an expertise of two research fields, the computer science and cognitive science domains. By employing the cross-domain methodology, we aim to demonstrate the key mechanisms, which eventually allow us to effectively unlock the proactive computing potentials. A fulfilment of the given objective may be eventually accomplished through a combined application of corresponding methods and approaches, issued from both research fields. An expertise of the cognitive science domain will successively complete the methodological needs of proactive computing research. The main composite aspects of our methodological structure are based on the specific multidisciplinary techniques from both research fields. Consequently, the research problem, outlined in the upcoming section is inherently linked to underlying empirical and methodological factors of the proactive system's framework.

1.3 Study objectives and key research directives

Proactive computing as a new research paradigm does not possess yet a clear methodological support and approach repository, which may successively provide us with the necessary techniques, methods, and tools for building a fully functional proactive system. Being a comparatively new research domain, which is still in its earlier stages of development, proactive computing necessitates a thorough empirical and theoretical investigation. The applied research efforts will consequently allow us to reveal and to formalise the approach's underlying functional characteristics and its associated methodological traits. The main objective of the current research project is to design, build, and test the coherent methodological and empirical frameworks, which will subsequently allow us to provide our proactive system with a scientifically rationalised proof of concept.

1.3.1 Research objectives and key methodological orientations

Basing on the underlying factual traits of proactive computing research, indicating a lack of clear methodological support, we therefore aim to elaborate and to put into place our own repository of guiding methods and approaches. Consequently, we decide to investigate the proactive computing paradigm within the defining principles of two computer science approach orientations. The given research directions respectively include deterministic and probabilistic methodological principles. Subsequently, in order to provide the supporting basis for all further system's conceptual and technical expansions, we define the corresponding functional principles for each applied methodological framework.

Thus, on a deterministic level, we specify the according functional specifications of the proactive system, which primarily stipulate that we have to anticipate a target context event, basing on the techniques of pre-programmed proactive scenarios and rules [10]. An application of the given principles has the initial objective to provide the stable system's functioning, with the first prerequisites of a proactive computing approach. Correspondingly, on the probabilistic level, we specify that the system has to anticipate a new event or to refine the context details of an initial event, basing on a probabilistic analysis of already available context information, which is originally provided by the functions of related deterministic rules [11]. Besides the mentioned event monitoring processes, the system has to additionally identify an exact activation time for the potential response-actions, related to a detected context event.

It should be noted, that according to the aforementioned methodological specifications, the implemented mechanisms of a deterministic approach provide the corresponding mechanisms of a probabilistic approach with the necessary functional prerequisites. In other words, the stochastic mechanisms of our probabilistic framework are conceptually linked to, and based on a functional output of the elaborated deterministic framework. More specifically, in order to refine the prevailing context details, and thus to penetrate into more subtle abstract layers of context characteristics, we need to identify first the initial prerequisites of a target event. An identified and collected context data must be an exact representation of an initial target event.

The given approach allows us to provide the corresponding system's algorithms with the event's original properties, necessary for all further probabilistic evaluations of context details. Consequently, in order to detect the predetermined target events, the associated context monitoring mechanisms of the proactive system have to be based on the underlying principles of a deterministic approach. On the other hand, in order to be able to penetrate into more subtle layers of prevailing context details, the system's analytical mechanisms must be correspondingly based on the underlying principles of a probabilistic approach. Our main objective for applying the aforementioned stochastic mechanisms is to be able to precise the quality and properties of the currently prevailing context characteristics.

Consequently, in order to fulfil the thesis' objective, we have to validate or, in other words, to formalise the theoretical and empirical specifications of both methodological frameworks. Notably, we have to formalise the proactive system's general structure, its underlying methodological framework, and eventually the quality of its primary functional capacities. The given approach will subsequently allow us to provide the necessary proof of concept for all applied concept designs and their corresponding technical realisations.

Due to the thesis' main objective, which stipulates the necessity to scientifically rationalise all proactive system's underlying mechanisms, we have to design, formalise, and test both, the governing theoretical and empirical aspects of our entire methodological framework. The given objective consequently leads us towards a specification of the thesis' research problem and all underlying research questions, which collectively define the structure of the current work.

1.3.2 Research problem and fundamental conceptual orientations

As we conceptually base our system on a new research dimension of the proactive computing paradigm, our proactive system does not possess a clear empirical and methodological support. Throughout our literature research, we have identified a lack of coherent specifications of the domain approach repository, and essential methodological definitions, which consequently implied that proactive computing research is still on its earlier stages of development. Therefore, for the purposes of an effective formalisation of the system's applied methods and approaches, we need a more extensive investigation on the theoretical and empirical levels. Consequently, the given approach allows us to provide the valid proof of concept for all designed conceptual elements and implemented functional characteristics of our proactive system. The given direction ultimately represents our main research problem.

In order to address the aforementioned research problem, and thus to provide the valid proof of concept for our proactive system, we subsequently formulate a set of research questions, which initially aims to guide us throughout the progress of our research. Notably, the stated research questions define the main directions of our study. Each formulated research question aims to address a specific aspect of the identified research problem. Collectively, all research questions define three main orientations of the applied methodological framework, including a theoretical basis, approach directions on the deterministic level, and approach directions on the corresponding probabilistic level.

Research question #1. Our first research question aims to determine the according theoretical directions and methodological prerequisites for all further specifications of applied methods and approaches. Thus, the Research Question #1 is formulated as follows:

> *What are the main theoretical and methodological prerequisites and orientations, which define the conceptual directions of a system, built within the framework of the proactive computing paradigm?*

The outlined research question helps us to specify the composite characteristics of the proactive system's conceptual directions on the underlying theoretical level. The initial theoretical specifications might include the methodology and expertise of several related computer science research fields. Besides identifying the main theoretical prerequisites, the given research question additionally helps us to define the system's conceptual orientations on the fundamental methodological level.

Research question #2. Our second research question aims to specify the main conceptual characteristics and design orientations, which subsequently govern the proactive system's development process within the fundamental deterministic paradigm. The Research Question #2 is formulated as follows:

> *What are the key principles and elements, which define the main system concept, its functionality, and its integrity on the fundamental deterministic level?*

Correspondingly, the stated second research question helps us to specify the system's main design orientations and certain functional attributes on the primary deterministic level. As we will see in section 3.2, a prospective list of underlying factors will include several concept implementations, which define the system's functional aspects on the fundamental deterministic level. The given research question provides a basis for a successive validation of the deterministic side of the system, which is primarily achieved through our first empirical study. The main objective of the aforementioned experiment is essentially characterised by an assessment process of relevance, quality, and outward implications of the system's functional attributes.

Research question #3. Our last research question aims to determine the main conceptual characteristics and design orientations, which correspondingly specify the proactive system's transition to a dimension of the probabilistic approach. The Research Question #3 is formulated as follows:

> *What are the key principles and approaches, which define the transition of the framework concept from deterministic towards probabilistic dimension?*

Our last research question helps us to specify the system's main design orientations and certain functional attributes on the secondary probabilistic level. As we will see in section 4.1, the prospective list of underlying factors will include several concept implementations, which accordingly define the system's functional aspects on a more intricate probabilistic level. The given research question provides the basis for the successive validation of the probabilistic side of the system, which is achieved through our second empirical study. The main objective of the aforementioned experiment is characterised by an assessment process of applied methodological techniques and approaches, which consequently aim to demonstrate the feasibility of concept realisation within the underlying principles of the probabilistic framework.

As we have mentioned earlier, the proactive computing paradigm is essentially characterised as a comparatively new field of the computer science research, which does not possess yet the clear methodological support and coherent approach repository. The given disposition implies that the current state of proactive computing research can not provide us yet with the necessary techniques, methods, and tools, allowing us to build the fully functional proactive system. Ultimately, in order to be able to build a system, incorporating proactive computing principles, we need to elaborate our own methods and techniques, which in the end will constitute the main body of our methodology.

Correspondingly, the underlying methodological structure of our proactive system is based on a series of theoretical and empirical studies, which aim to specify the compound characteristics of the framework's functional and theoretical aspects. The aforementioned series of studies are consequently divided into composite research steps, which are initially based on several guiding principles, specified in form of our earlier presented research questions. Ultimately, each research question aims to stipulate a specific factor of our methodological framework.

Our first research question aims to emphasise the need for a coherent theoretical basis, which will be used as the starting point and the supporting ground for all further elaborations of the proactive system's aspects and their underlying functional mechanisms. Our second and third research questions, consequently aim to help us in the specification and definition of the methodological key orientations and governing characteristics of the system's conceptual attributes. Additionally, for an effective elaboration of the system's proof of concept, the last two research questions aim to stipulate that we will need to test and to validate the chosen methods and approaches through the corresponding empirical studies. Thus, a formalisation process of the system's conceptual aspects and their underlying methodological structure is consequently achieved in a progressive manner through the design, elaboration, implementation, and ultimately testing and validation of all system's composite aspects.

1.4 An outline of the key methodological aspects

Below, we highlight the successive methodological steps, which accordingly describe all essential theoretical and empirical aspects of the current thesis. In the given section we outline the project's key methodological factors, which collectively constitute the main body of the thesis. Consequently, the elaborated methodological framework represents the underlying structure of our proactive system and its composite conceptual and empirical factors.

1.4.1 Framework formalisation

The key methodological factors of the current study are essentially defined by a formalisation process of the chosen concepts and methods within the underlying principles of two theoretical perspectives. The designated approach orientation is necessary as it allows us to build a required proof of concept in the progressive interrelated structure. More precisely, we design, elaborate, and implement our methods and the system's primary functional capacity by following the principles of the deterministic approach. The given research specifications consequently provide us with the possibility to build first the stable fundamental structure of the proactive system. The elaborated fundamental basis allows us to subsequently implement more evolved context monitoring and data analytical techniques. We design and implement the corresponding probabilistic part, conceptually basing on the formalised mechanisms of the system's deterministic framework. In other word, by applying the probabilistic computational techniques, we try to extend the system's initial context monitoring capabilities, and thus to increase the accurateness and intuitiveness level of successive proactive mediation. Therefore, an effective formalisation of the fundamental deterministic mechanisms allows us first to assess the functionality and potentials of the key principles of the basic proactive functions and their successive outward impact. On the other hand, an effective formalisation of the system's probabilistic mechanisms allows us to go further, and thus to explore the fundamental principles of the proactive computing paradigm on more advanced levels.

Consequently, a conceptual realisation of the system's deterministic side is essentially based on the design and application of several computational techniques and methods, which provide the system with the basic functional capacity. In order to implement the underlying proactive computing principles, we design a number of the system's initial deterministic components, including the Rules Running System, proactive rule and its fixed algorithmic structure, proactive scenarios and their inherent task delegation mechanisms [5, 12].

For the purposes of the context monitoring function, we design several proactive scenarios, which have the objective to continuously scan a target context environment for the predefined instances of corresponding context conditions. Ultimately, the given approach allows the system to initiate, if needed, the successive context mediating actions.

Conceptually, a proactive scenario is represented as a set of compound predefined rules, which specify the scenario's main type and its basic functional characteristics. The allocated predefined rules are consequently executed by our Rules Running System, which conventionally ensures the continuity aspect of the proactive system's monitoring and mediating processes. Therefore, all aforementioned composite elements, built within the principles of the deterministic framework, collectively provide the stable operational basis for all further expansions and implementations of the system's new functional aspects.

An effective implementation of the stable deterministic framework successively allows us to design and to implement more elaborate context monitoring and context mediating techniques, based on the principles of the probabilistic approach. Conventionally, the system's probabilistic mechanisms include several composite modules, which collectively allow us to refine the accurateness of the system's monitoring capabilities and to increase the intuitiveness level of the system's mediating actions. Thus, the new framework incorporates a function of probabilistic data evaluation through an application of the associated techniques of the Bayesian approach. Additionally, the given framework includes an adaptation of corresponding interdisciplinary techniques, initially borrowed from the cognitive modelling approach [11]. Subsequently, an application of both approaches allows us to refine the system's functional capacity by increasing the intuitiveness and accuracy levels of its monitoring and mediating techniques.

Consequently, the formalisation of both methodological approaches plays a crucial role in an overall objective of providing our system with a valid proof of concept. More specifically, an effective validation of chosen concepts, theories and methods, collectively allow us to provide the scientific evidence, necessary for a well grounded proof of concept. Conventionally, a methodological structure of concept validation is represented as a gradual formalisation process of all system's aspects. First, we aim to elaborate, design and formalise the system's fundamental mechanisms, basing on the underlying principles of the initial deterministic approach. Subsequently, we aim to elaborate and to formalise the successive stochastic mechanisms of context data evaluation, basing on the underlying principles of the associated probabilistic approach. Therefore, an effective validation of both methodological frameworks and their corresponding computational techniques will provide our system with the necessary, scientifically rationalised proof of concept.

1.4.2 Preview of concept realisation

A formalisation of the system's concept consequently necessitates an elaboration of applicable methods and approaches, which will allow us to incorporate the underlying functional principles of prospective methodological frameworks. As we will see in our upcoming chapters, one of the fundamental elements of the proactive system is a proactive scenario. The underlying mechanisms of all proactive scenarios ultimately provide our system with the valuable proactive behaviour. Consequently, a variation of designed and implemented proactive scenarios defines the orientation of the system's functional characteristics. Therefore, in order to build the proactive system, which is conceptually based on two distinct methodological frameworks, we have to correspondingly design two distinct variations of the system's proactive scenarios.

An ultimate formalisation of both methodological frameworks is obtained through the design, implementation and testing of two waves of proactive scenarios. Thus, the main design directions and conceptual characteristics of the first-wave proactive scenarios aim to define the fundamental algorithmic mechanisms, which will consequently allow us to implement the basic functional capacity of the system's proactive behaviour. On the other hand, through a formalisation of underlying mechanisms of the first-wave scenarios, we are able to create the stable functional framework for all future types of proactive scenarios. Therefore, the first-wave proactive scenarios are essentially characterised as a fundamental realisation of the initial concept, which is designed and built within the governing principles of an elementary deterministic approach.

Correspondingly, the main design directions of the second-wave proactive scenarios aim to define more accurate algorithmic mechanisms, which will consequently allow us to increase the intuitiveness and accuracy levels of the system's monitoring and mediating capabilities. Conceptually, the given set of proactive scenarios is based on, and designed according to the functional principles of the first-wave proactive scenarios. However, an implementation of additional probabilistic and cognitive modelling techniques distinguishes both types of proactive scenarios in terms of their complexity, functional capacity, and acquired qualities of the system's monitoring and mediating capabilities.

1.4.3 Experiments' key orientations

As we will see in our later chapters of theoretical and methodological descriptions, a formalisation process of the proactive system's key mechanisms may be consequently achieved through dedicated empirical studies. The prospective experiments will aim to investigate the various aspects of the system's implemented methods, techniques and approaches. Both waves of proactive scenarios, together with their underlying functional mechanisms will be therefore subjected to the thorough investigation, performed within the framework of two experiments.

The key objectives of the first experiment will consist of organising the dedicated *enquiry study*. The given experiment aims to test the system's fundamental functions and their overall performance, to examine the system's functional capacity and to study an impact of the system's proactive behaviour on a user's general performance. In other words, the first empirical study aims to validate the deterministic part of the proactive system, and on the other hand, to determine the scale of its functional impact on a user. Ultimately, an effective formalisation of deterministic mechanisms will open the doors for a successive implementation of supplementary techniques of the probabilistic approach.

The key objectives of the second experiment will consist of organising the corresponding *exploratory study*. The given experiment aims to elaborate and to enhance the system's initial functional capabilities by applying more advanced techniques of the probabilistic approach and cognitive modelling methodology. In other words, the second empirical study aims to validate the probabilistic part of the proactive system, and on the other hand, to emphasise the *feasibility* of a concept realisation. Moreover, the mentioned exploratory study has the objective to highlight the value of a multidisciplinary methodological approach. Ultimately, from a theoretical point of view the results of the second experiment will help us to outline potential directions for future research efforts.

1.5 Thesis structure

Before proceeding further towards the detailed description of our enquiry and exploratory studies, we provide in the present section the main guidelines for our thesis structure. Conceptually, the thesis organisation is divided into three thematic parts, where each part successively describes a specific aspect of the study.

Thus, the first part is characterised as an introductory section of our work, which initially confines two chapters, *Introduction* and *Theoretical framework*. Both chapters aim to introduce to the reader the main study concepts, theories, and definitions, which are successively used in the following chapters. Chapter 2 of the literature review addresses the underlying aspects of our Research Question #1, which aims to specify the conceptual and theoretical characteristics of our interdisciplinary methodological framework. The given chapter has the objective to situate the study within a clearly defined theoretical structure. Here, we introduce the key concept of *Proactive computing* together with other related theories and approaches, which are conceptually linked to the specifications of our methodological framework.

Our second part represents the cornerstone of our thesis, which contains two main chapters, *Deterministic methodological framework* and *Probabilistic methodological framework*. Both chapters aim to delineate the underlying methodological aspects of a concept formalisation. Chapter 3 addresses the underlying aspects of our Research Question #2, which aims to highlight on an empirical level a *deterministic side* of the proactive system. Additionally, in the given chapter, we specify the settings of our first empirical study, which aims to investigate the functional capacity of our initial deterministic framework.

Consequently, our chapter 4 addresses the underlying aspects of our Research Question #3, which aims to highlight on an empirical level a *probabilistic side* of the proactive system. In the given chapter, we specify all conceptual and methodological characteristics of our system, which collectively aim to demonstrate the *feasibility* of the concept realisation within the principles of the probabilistic approach.

Our last part represents the conclusive component of our work, which initially confines two chapter, *Study results and general discussion* and *Conclusions and future directions*. Both chapters aim to conclude the current work by highlighting the nature of obtained results and emphasising the characteristics of their general implications. Chapter 5 delineates the major study findings, their implications, and relationship towards the applied theory, methods, and approaches. Additionally, in section 5.2, we unify in a discursive form the main points of the study by highlighting the key relational aspects of the experiments' results with regard to our initially stipulated research objectives. Consequently, chapter 6 concludes the work with an extensive highlight of all fundamental study aspects. Additionally, we outline here our visions for the prospective research efforts.

Chapter 2

Theoretical framework

In the given chapter we present the main theoretical framework, which contributes to the understanding of key ideas, definitions, and approaches, used in the current study. The highlighted theories, presented in the upcoming sections collectively address the underlying aspects of our Research Question #1, which aims to specify the conceptual and theoretical characteristics of our interdisciplinary methodological framework. The given chapter has an objective to situate the study within the clearly defined theoretical structure. Therefore, all successively used definitions, notions and methodological orientations are based on, and related to the delineated below theories, approaches and methods.

The given chapter is conventionally divided in two theoretical perspectives of computer science and cognitive science research. In section 2.1, we introduce to a reader the associated concepts and research orientations, related to the computer science domain. First, we provide the definition and corresponding functional characteristics of proactive computing paradigm, which is used in our study as the underlying basis for all methodological and empirical aspects of the system's design. Additionally, we acquaint a reader with all associated computer science approaches and theoretical orientations, which have been applied or referenced throughout the current study. The mentioned theoretical orientations include *Ubiquitous computing*, *Autonomic computing* and *Context-aware computing*. Ultimately, in section 2.2, we highlight all corresponding theories and approaches, related to the cognitive science domain. Our objective here is to introduce to a reader all applied theoretical and methodological aspects of the *Cognitive psychology* and *Cognitive modelling* approach. Consequently, in the end of the given chapter, we highlight an interdisciplinary nature of the current research, where we emphasise the necessity of applying the cognitive science expertise together with a repository of its methods and approaches.

2.1 Computer science perspective

The current section gives an overview of several computer science theories and methodologies, which make a theoretical basis of the given thesis. The presented below approaches and theoretical orientations aim to assist a reader in a better understanding of applied methodologies, used in the current work. All highlighted theories are delineated in the logical sequential structure, which has an ultimate objective to progressively lead a reader throughout the study's central points, representing the basis of the current methodological framework.

2.1.1 Proactive computing

Conceptually, the architecture of our software system is based on the notion of *proactivity*. The application of this approach defines the main directions of our theoretical framework. In order to provide a primary general definition, below we highlight several features and underlying conceptual characteristics, pertinent to the design specifications of any proactive system type. Thus, a system, which employs the concept of proactivity is primarily characterised by the following inherent capabilities:

- continuous awareness of surrounding context conditions,
- intuitive monitoring of a related context evolution, and
- functional capacity to generate the dedicated proactive actions with regard to the needs of a context situation [13, 14].

On the one hand, a system, enhanced by specific proactive features is capable to act on its own initiative by continuously providing the adapted proactive services. On the other hand, a system amplified by proactive type of behaviour is capable to be perceptive and intuitive towards the manifested, context-based needs of a user. The aforementioned inherent characteristic of a proactive-based system, consequently prompt us to use the given approach as main conceptual framework of our system.

It should be noted that in our research, we understand and use the notion of *Proactive Computing* as it is initially defined by Dr. David L. Tennenhouse [1, 2]. In his vision of the future computer science research he explores a new paradigm, where he proposes (a) to re-examine the relationships between physical and abstract domains of a context, (b) to re-evaluate our general approach of interactive computing, and (c) to gradually move from interactive human-centred to proactive human-supervised models of human-computer interactions. By taking into consideration the aforementioned factors, we are therefore able to build more intuitive, user-oriented computer technologies.

According to the definition of the proactive computing, given by David Tennenhouse, there are two main principles, which govern the entire process. The first principle stipulates that any proactive system is always working on behalf of a user, whereas the second principle specifies that a proactive system acts on its own initiative without explicit instructions from the user [1]. In other words, we can assume that the system may possess a set of policies, which define the patterns of its behaviour in various contextual situations. Therefore, as a general characteristic, in order to prevent an unwanted event from happening, a proactive system always aims to cause a change in a context, rather than just to react to changes.

In order for the system to be able to monitor the outside world, it has to possess some instruments, allowing to be aware of a surrounding situation, and to be able to capture certain aspects of a context. In his work, David Tennenhouse outlines the prerequisites for such a mechanism by highlighting the necessity of implementing different sensors and actuators, which will serve as the perception centre of the system. Such a mechanism will allow the system to interact with the world around it using its sensors, and thus to capture and to monitor an event of interest.

The process of proactive mediation itself is characterised by several features. The most distinguishable property of a proactive system is its capacity to act according to the needs of an evolving situation on its own initiative by providing the adapted services [13, 14]. Another important characteristic, which distinguishes the proactive system, is its capability to act with respect to future possible situations. Thus, during the process of a context monitoring, the system can translate the attributes of a currently prevailing situation together with the contextualised user's actions, and use it as an evidence for future event estimation [6, 7, 15].

In the prospect of his vision for the future of computer science research, the prevailing motivation for David Tennenhouse to move from interactive human-centred towards proactive human-supervised computing seems to be stipulated by an ever-growing number of networked computers. The given disposition presumes and necessitates the review of current techniques and mechanisms for elaborating more adaptive proactive processes of interaction between networked computers with respect to the specifics of a context and the needs of a user. Thus, David Tennenhouse proposed three guiding principles in the efforts to move to a new paradigm.

The first principle is defined by the condition of connecting the proactive system to the world around it, using sensors and actuators in order to acquire the context-aware capabilities for monitoring the physical domain. At this level, the system will be capable to capture the events of interest for its further analysis.

The second principle is partially dependent on the first principle in terms of data evaluation and further data exchange between the networked embedded processors. Such data exchange, according to Tennenhouse, has to function at a higher operational frequency than humans could possibly sustain, which simply implies that the latency between system inputs and outputs are considerably shorter. In other words, in order to communicate with a user within the interactive computing paradigm, the system's operational frequencies must always be adapted to the frequencies of a human response. In the proactive computing, the system's operational frequencies are much faster, which puts automatically a human agent outside of the interactional loop. In this perspective, the computing processes of a proactive system will have a faster response rate to an external stimuli in comparison to a situation where human decision making is involved. Therefore, according to Tennenhouse, the proactive computing will not be characterised as human-centred but human-supervised, meaning that the allowed extent of a decision making will be done mainly by the system but under the constant supervision of a human agent.

This leads to the third principle, which specifies the naturally derived condition of placing human agents outside of the interactive loop. The given disposition, as we mentioned earlier is due to the shorter time constants in proactive system interactions with the associated data of an outside context. According to David Tennenhouse, the human involvement will shift from direct human-computer interactive tasks into supervisory and policy-making tasks. The systems, therefore, will not be in direct contact with human agents, but rather with their environments for the objective of providing proactive actions on its own initiative where it is most needed. In order to implement certain attributes of the proactive computing, Tennenhouse highlights in his work the necessity of employing in a system design the probability approach, and thus, gradually moving from deterministic to probabilistic models of computation [1, 2].

Proactive computing is therefore a new paradigm, which according to Tennenhouse lies within one of the domains of *Ubiquitous computing*. Furthermore, the author in his attempt to ground a proactive computing approach, divides the methodological scope of ubiquitous computing into several spaces. On the one hand he specifies the dimension of interactive, manual, human-centred computing, which is more common in the traditional sense of the term. On the other hand he introduces the dimension of proactive, autonomous, human-supervised computing, which according to Tennenhouse necessitates more extensive research efforts in order to expand the common trends in computer science research beyond the interactive domain. Thus, the first dimension designates more specifically the reactive character of human-computer or machine-to-machine interactions with a human agent in the main loop, whereas the second dimension specifies the proactive character of these interactions by giving to a human agent the supervisory role.

We may notice that an initial idea of proactive computing as it was defined by Tennenhouse, lies within the same methodological space as ubiquitous computing. The author places his new envisioned concept into a not yet fully developed dimension of ubiquitous computing, and thus encourages the researchers to take this approach further for its development and advancement. By taking this directive into consideration, we therefore aim to investigate the nature of proactive computing for its better understanding by analysing more extensively the main aspects of ubiquitous computing. Our objective in this analysis is to find the potential common points as well as individual unique characteristics, which distinguish the two aforementioned concepts.

2.1.2 Ubiquitous computing

As a branch of computer science research, *ubiquitous* or sometimes called *pervasive computing* has originated more than a decade ago. The notion and concept have been introduced by Mark Weiser in 1991 [16]. In his paper, he projected the ideas of pervasive or ubiquitous computing, which he described as invisible embedded systems, surrounding and helping people in the tasks of their everyday life. In his definition of ubiquitous computing, Mark Weiser states the idea that the technology becomes profound only when it disappears or becomes indistinguishable in people's everyday life. Such "disappearance" of technologies, according to Weiser, is due to their seamless integration into user's environment. He links the mentioned "disappearing effect" to the human psychological state, where through the perfect learning of something, a person starts to gradually loose his/her awareness about the subject of learning due to its complete internalisation by the mind. The same happens with the technology, if it is seamlessly integrated into the background of people's everyday environment.

In order for technology to be ubiquitous, it has to be designed and developed around the idea of seamless and undetectable background functioning. Russell et al. highlights in his work the importance of accurate system design and its integration into the environment by specifying the guidelines, which he designates as *Heterogeneity*, *Dynamism*, *Robustness*, and *Interaction techniques* [17]. In summary, the aforementioned aspects highlight (a) the importance of diversity of embedded systems in a workspace, (b) the dynamic nature of a software framework, allowing to easily update the embedded modules, (c) the general system stability, allowing to provide a non-interrupting functioning, and (d) the adapted interaction techniques, allowing to provide adequate and efficient interfaces.

Thus, within the aforementioned definitions of ubiquitous computing, we may highlight one important aspect, which constitutes the basis of this type of computing paradigm. The environment in this case represents the key factor, which defines the term *Ubiquitous*. In his work, Lupiana et al. tries to categorise the ubiquitous environment in two major classes [18]. The authors define *Interactive* and *Smart* environments, where the former represents the setting for a group work, and the latter represents the setting for an individual work. In the case of *interactive environments*, the embedded computer systems are often playing the role of supportive work tools without any smart capabilities. The *smart environments* on the other hand, according to Lupiana are the supportive embedded systems, which assist people in their everyday life. The main objective of ubiquitous computing technologies is therefore to provide enhanced living and working environments.

The functional capacity of the given technology is mainly based on the idea of tracking the user's current location and some other contextual data, which is acquired through wireless devices and signal transmitters [19]. Such systems are usually programmed to observe an event and to react to the given event automatically with the objective either to provide an interactive feedback, or a designated service. We may notice here, some prerequisites of a proactive dimension that has been mentioned by Tennenhouse, however the degree of its realisation and implementation, in our opinion, is mostly characterised as interactive.

In another work, highlighting the visions and challenges of ubiquitous computing, the author shows through hypothetical context scenarios that the proactive type of system behaviour is indeed a missing link in pervasive computing [20]. The author states that it is crucial for a pervasive system to be able to track user's intent in order to determine the appropriate system actions with the objective of helping a user in his/her task activity. Therefore, the author stipulates the importance of anticipation of user's intent for providing the adapted and proactive support.

In his work on expert systems within the framework of ubiquitous computing, Kwon and his colleagues argue that most of the current ubiquitous computing-based applications provide only limited personalisation services using user's contextual data. Furthermore, the author states that an expert system within ubiquitous computing space has to be reinforced by a proactive type of behaviour in order to be able to provide adapted and intelligent decision making capabilities [19]. According to the author, one of the main reasons for applying *Proactiveness* is the necessity to have the self-triggering mechanism, allowing to infer the user's context-based needs, and thus to provide a better service quality. Consequently, the author lists proactive capabilities as one of the main prerequisites, essential in the next-generation, ubiquitous, decision-support systems. Due to the conceptual characteristics, which stipulate that a system, operating within the user's environment is always context-dependent, the author additionally highlights the importance of context-aware capabilities.

We see the same paradigm of indispensability of context-aware capabilities yet in another example. In his attempt to define the main conditions for a pervasive computing, the author stipulates that in order for a system to be able to accordingly adapt its behaviour it has to be cognisant of user's states and surrounding context conditions [20].

In another work, related to system adaptation research, the author highlights the necessity for pervasive computing applications to be able to self-adapt to the changes of a target environment in order to maintain the up-to-date state of their interactions with the outside world [21]. The author puts an emphasis on self-organising systems within the ubiquitous space, where he compares two types of adaptation, *reactive* and *proactive*. Furthermore, the author states that in the first case of *reactive adaptation* the frequent context change leads only to the frequent system adaptation, which doesn't seem to be always suitable. On the other hand, in the second case, *proactive adaptation* allows the system through anticipatory features to calculate ahead the possible or necessary configurations, and thus to optimise system's decisions. In order to decrease the time adaptation, the proactive features are used for anticipation and prediction of certain context attributes and user's general intent. The author designates the given approach as *proactive adaptation*, which allows the system to promptly anticipate the changes within an associated context environment, and to accordingly adapt its behaviour.

Through the aforementioned examples we may notice the multiple sides and subdomains of ubiquitous computing. This demonstrates that from its conception in 1991, the notion and the field itself grows exponentially and evolves towards new approaches and techniques. The given disposition may serve as a prerequisite for triggering and encouraging the research of new computing paradigms. In his analytical paper, highlighting the aspects of proactivity in service-based applications, Vlahakis et al. argues that with the spread of computing power and ever increasing rate of computer devices, consumer electronics and embedded micro processors, the ubiquitous computing provides indeed a stable ground for a gradually developing and growing methodology of proactive computing research [13].

In the brief chronology of ubiquitous computing, highlighted above, starting from its conception in 1991, we may notice a slight shift in computer science research towards the related proactive dimension, discussed by David Tennenhouse in his work. The given shift, consequently demonstrates the increasing necessity for applying the additional research efforts, allowing to develop an effective methodology for the corresponding proactive mechanisms.

In our opinion, one of the possible preconditions for such a shift towards the dimension of proactive computing may be characterised by the continuous expansion and evolution of ubiquitous computing research and its technologies. Consequently, the given evolutionary aspect marks the progression of processing capabilities and increasing rate of networked embedded processors. The gradual evolution of computing technologies emphasises the emerging need for new solutions, which will successively allow us to manage the ever growing computing complexity. One of the possible solutions, which aims to address the aforementioned problem has been proposed by the researchers at IBM with the formation of *Autonomic computing*. However, the underlying concept of autonomic computing partially overlaps the proposed framework of proactive computing from Intel Research. In the next section we will try to analyse the specifics of both fields and to highlight their most notable differences and similarities.

2.1.3 Autonomic computing

Within today's society, characterised by an ever increasing rate of use of different technologies, the computer systems became more complex and more elaborate in relation to their operation and management. Therefore, the emerging need for creating an optimal solution, allowing to tackle the forthcoming challenges of managing the ever-growing system complexity became more obvious. As a result, in 2001, the researchers from IBM proposed in their manifesto a new approach of *Autonomic computing*, which aimed to address the impending issue [22]. The main aspect of autonomic computing can be characterised by the system's ability for self-management and adaptation, according to factors of its context environment [23]. The main points, which characterise the new approach have been specified by researchers. In order for a system to be able to perform self-management functions, it has to possess four properties: self-configuration, self-optimisation, self-healing, and self-protection [22, 24].

Besides these four attributes, an autonomic system has to possess as well the various quality requirements, where one of the elements is an anticipatory feature [25]. Therefore, according to the characteristics of an autonomic system, we may notice a certain overlap with the proactive computing approach. The following example points to these correlations between two aforementioned subfields of computer science research. In a work, which analyses the autonomic principles within cloud computing, Martinovic et al. highlights the general need for predictive or anticipatory features [26]. According to the author, in a system, which operates within e-health environment, the error resiliency may be built on anticipatory features based on a proactive type of system behaviour. In order to clarify the common points and to distinguish the unique characteristics we present below a comparative overview of the two approaches.

In our opinion, the most comprehensive and complete comparison of two research directions has been done by researchers of Intel themselves, that is Roy Want, Trevor Pering and David Tennenhouse. We find such overview extremely interesting and valuable for our work, which consequently allows us to see the main points and arguments from the perspective of proactive computing founders themselves.

The authors start by underlining the main principles of autonomic computing, which constitute the cornerstone of its structure. The given principles can be characterised as follows:

- an autonomic system has to be capable to self-monitor, self-heal, self-configure, and self-improve its performance,
- the system has to be capable of context-awareness and self-defence against external attacks,
- the system has to be able to interact, and furthermore
- the system has to possess some anticipatory features [2, 22].

On the other hand, the authors highlight the main principles of proactive computing, which can be correspondingly specified as follows:

- a proactive system has to be connected to the outside world or any other associated context environment,
- the system has to be capable for deep networking, and macro-processing,
- the system has to be able to anticipate, and to deal with uncertainty,
- the system has to be the closing element in control loop, and
- the system has to be able to provide adapted and personalised services.

In his further analysis, Want et al. highlights the common advantage of context-awareness for both systems, either in configuration support for an autonomic system, or in adaptation of services for a proactive system. In several examples presented in the paper, the author highlights the common, mutual benefit of two fields, where for instance, the principles of autonomic computing are playing an essential role of inter-connecting the various components of a system, or providing the self-defence capabilities against external attacks, or simply unwanted data or program. On the other hand, the principles of proactive computing are crucial in a general system management and coordination, which are mainly based on system's anticipatory features. Thus, the authors put both research perspectives, and their conventions as the emerging necessity for all scalable systems. The main argument made by Want et al. is that both autonomic and proactive computing principles will most probably be the guiding factors in the upcoming computer science research.

According to the aforementioned theoretical highlight, the conceptual framework of autonomic computing contains some properties, which are equally common to proactive computing paradigm. However, we may notice that the key difference between the two fields becomes more apparent in the area of its application. That is, an autonomic system seems to be more oriented towards its internal operation and self-management, whereas the proactive system is more dedicated towards the user-related context activity. According to the conceptual highlight, made by Want et al., in conjunction, the two computing approaches complete each other, if integrated within one methodological framework. In other words, a system, which uses both concepts, acquires internal autonomic self-management capabilities and a context-oriented, external proactive behaviour. Therefore, the aforementioned theoretical review puts the two research directions onto one line as parallel, complementary approaches, rather than conflicting and opposing concepts.

Throughout the entire chapter of theoretical framework up to this point, we may notice an existence of a unique element, which is uniformly present in all highlighted theoretical approaches of proactive computing, ubiquitous computing, and autonomic computing. Namely, the given element is *Context-awareness*, which seems to be shared by all mentioned theories due to their specific needs to be aware either on the internal low-level of some system aspects, or on the external high-level of user-oriented context situations. The necessity of connection to external or internal context attributes is primordial for interactive context-based human-computer interactions in case of ubiquitous computing, or for internal system-based self-management in case of autonomic computing, or for proactive context-based user guiding in case of proactive computing. Thus, in the next section we investigate further the field of context-awareness in the frame of proactive system research in order to understand its main theoretical and conceptual underpinnings.

2.1.4 Context-aware computing

Due to the progressive development of computing technologies and their methodologies, it is fair to imagine that *Context-aware computing* most probably appeared as a logical consequence of the arrival of pervasive technologies on the scene of computer science research. In this perspective, the evolution of ubiquitous computing may be considered as a first precondition for the design and development of the context-aware systems and their approaches. Therefore, in our further analysis we try to define the notion of *context-aware computing*, to distinguish its strong and weak points, and we try to analyse its role within the framework of proactive computing research.

In order to define and to understand the term of *context-aware computing* as an approach within the field of ubiquitous computing, we must examine it independently as a separate concept. The beginning of context-aware computing as an approach started in the early 1990s with the introduction of the first context-aware application, which operated within an office environment [27]. In his work, Want et al. introduced a novel approach of locating a person by using an electronic badge, which was transmitting a signal in form of a unique code. The emitted signal then had to be captured by a network of sensors in order to be used later as a beacon of a person's location [28]. This *Active Badge Location System* was the first system to be used within the framework of the generally established notion of context-awareness.

However, the term of context-aware computing has been introduced few years later by Bill N. Schilit and Marvin M. Theimer in their paper about mobile distributed computing [29]. The authors describe context-aware computing as an ability of a system to discover and react to changes of an associated context environment. The monitoring of a context itself consisted of collecting the location data in relation to objects of interest and their various states. The monitoring process included as well the detection of nearby people, nearby devices, person's identities, and some location-specific information such as electronic messages. Throughout the succeeding years of computer science research, the context-awareness as a notion started to expand and acquire new meanings, and thus forming its current-state definition.

Notably, in his work related to a context-aware archaeological assistant, Ryan et al. gives the definition to a context-awareness as the system's capabilities to sense an environment information, including location, time, temperature and user identity [30]. In another example, Brown P. J. defines context-aware capabilities as a set of the following characteristics: detection of *location*, *adjacency of objects* within an environment where people wear electronic badges, *critical states* such as temperature level, *computer states*, *identities* of people, and *time* [31].

Consequently, under the light of the given examples, the evolving character of context-awareness' paradigm becomes more evident. That is, in comparison to the first mentioned definition of context-awareness, the notion starts to acquire new meanings and attributes, such as *computer states, objects' adjacency* and so on. In order to identify the constituent parts of a context-aware application, and to define a context itself, Dey et al. conducted a survey work, which aimed to develop a theoretical ground for a context-aware approach [32]. In his research, Dey et al. defines first, the type of a possible active participant within a context, or in other words, an entity type. Thus, according to authors, an entity could be a person, place, or an object, which is related to, or associated with the human-computer interactions. The context itself is then defined as any information that can characterise a situation or an environment of an entity. To put it differently, any type of information is an attribute of a context as soon as this information allows the system to specify a situation, related to an entity, that is person, place or an object. Therefore, we may notice the terminological shift of the initial definition of context-awareness towards a more generic and inclusive description of its notion and its constituent elements. In contrast to its former initial denotational state, the latter definition type adheres more closely to our view of context-awareness, applied within the framework of proactive computing research.

Due to their pervasive capabilities, context-aware systems may be used in a variety of domains, which highlight different aspects of a person's life and attributes of his/her context activity within a specific environmental setting. Thus, throughout its conceptual development, the approach of context-awareness has been applied in a variety of domains, which resulted in the formation of different types of context-aware systems. Below, we present some examples of context-aware system applications.

As a general tendency, the majority of context-aware systems are oriented towards providing situation-based personalised services, or in other words, recommendations. Considerable amount of context-aware systems are in fact recommendation systems, adapted to their particular context environments. In the following example, discussed by Hariri et al., we can see the authors' attempt to augment, or to extend the general purpose recommendation systems, based solely on a user's profile by incorporating additional contextual information, which helps to define the best recommendation match [33]. The authors build the recommendation features of their system on a unified probabilistic model, which aims to merge the user's profile and context data for the statistical analysis of recommendation elements. In their work, the authors adhere to the definition of a context, given by Dey et al., which defines the context as any data, capable of describing the entity's current situation [32]. This position allows the authors to employ all available contextual information, obtained throughout the user's interactions with a system, in order to further estimate its probabilistic value for the customised recommendations.

Context-awareness is indeed a valuable resource of information, allowing to incorporate various user-oriented services. The given approach has found a stable ground in the environment of e-health services. In the next example the authors use an association rule within a decision-making model in order to identify the characteristics of recommendation services, based on contextual data [34]. Consequently, the personalised exercise services are provided, basing on the detection of patient's preferences from context data. In a similar example, which shows the enhancement of a *Recommender system* by context-aware capabilities, the authors propose a latent probabilistic model for context-based recommendations [35]. In their work, the authors present a combinatorial framework of *Context-aware recommender system*, which takes into consideration the context conditions in order to effectively predict the user's preferences. The authors base their system on a latent probabilistic model, which is used to extract the user's preferences and the recommending items' features. The chosen approach allows the authors to model a range of functional relations between context attributes, user preferences, and items' characteristics.

In the last three examples, we may notice the implication of an additional element within the context-aware system, which is mainly engaged in computing the relevance level of a recommendation. Notably, this element is often characterised as a decision-making component, which is commonly based either on a probabilistic data analysis, or on the simple statistical derivations. Additionally, according to the examples mentioned above, the predictive features, encompassing the context attributes, or the user's preferences seem to play an important role in the general perspective of a context-aware approach. Hence, from its beginning as a simple concept of context's properties recognition, we may notice the gradual evolution of context-aware computing towards more complex context monitoring and data analytical systems. The key functional capabilities of such systems may include the features of context-based adaptation, context derivation, context prediction, or the probabilistic matching of context-based recommendations.

As it is generally the case with the evolutionary principles, where the growth and development often involves new challenges, new confrontations, and new issues, undoubtedly, the same principles adhere to the evolutionary stages of a new approach or methodology. In its early stages, the context-aware computing had certainly to cope with a lesser number of methodological and implementation-related issues comparing to nowadays features-enhanced, context-aware paradigm. In the following paragraphs we try to highlight the most significant challenges that context-aware computing confronts today.

Due to the default characteristics of any context-aware system, to continuously collect a situational data, the context conflicts within an evolving situation can therefore be denoted as the first-hand problem type [36]. In his paper, Filho et al. proposes a mechanism, which aims to reduce an error ratio of the context-aware adaptation decisions, based on the implementation of context-quality indicators [37]. The authors highlight the importance of a context consistency management in order to be able to effectively provide context-based services. To achieve the mentioned goal, the authors developed a *Context-quality management* module, which performs the quality check, and thus allows the system to improve the reliability of context management operations and the system's context-based adaptations.

Throughout the last few examples, we may notice one prerequisite of a context-aware system, which is often characterised as a self-adaptation feature. The given system function generally aims to adapt according to an evolving situation either the system behaviour or its services, which is usually characterised as a conceptual challenge [38]. Correspondingly, in his work Hussein et al. addresses the issue by highlighting the general problematic disposition of a research community of treating separately the self-adaptive and context-aware paradigms [39]. According to the authors, such approach represents a strategical problem, where on the one hand, the research on self-adaptive systems is mostly characterised by the implementation of alteration functions and adjustment features. On the other hand, the research on context-aware computing is more concerned by context modelling and context management, without taking in consideration the benefit of mutual relationships between two perspectives. Therefore, in the framework of our research, we adopt a holistic perspective on conceptualising and developing context-aware capabilities for our system.

Another important challenge, often highlighted within the context-aware community, is characterised by many researchers as the lack of predictive or anticipatory features [15]. Consequently, in a work, performed by researchers at the Kyunghee University, the authors highlight the predominant *non-proactive* standpoint of current context-aware computing research, which is mostly characterised by developing systems with limited personalised context-based services [40]. According to the authors, the current service-based technologies are not yet capable to fully implement the smart decision support due to insufficient research efforts in combining the systems' decision approaches and the ubiquitous computing technology. Thus, in their work, the authors apply an approach of incorporating the proactive system reasoning, based on user's contextual data within the framework of a *Decision Support System*.

In a similar example, the authors stipulate the importance of using historical context data, which allows the system to apply the associated learning algorithms with the objective to predict a certain contextual information for the successive proactive services [15, 27]. Together with the proactive computing methodology, an implementation of anticipatory capabilities furthermore necessitates expertise in translating the corresponding cognitive attributes of a user's behaviour into the system's algorithmic level.

Thereupon, due to its quality of pervasiveness, a context-aware system usually entails different aspects of a context, including a human factor. Indeed, one of the main principles of such system is to be aware of user's contextual settings, that is location, preferences, and user's intentions, which accordingly touch the cognitive side of the approach. For this reason, a human factor represents a substantial challenge in conceptualising and developing context-aware system behaviour.

The given issue has been also addressed by other researchers in the domain. Notably, Verbert et al. discusses the issue of using Technology Enhanced Learning paradigm within the framework of recommender systems [41]. However, an effective implementation of the context-awareness in regard to a user's learning activity, evokes a lot of discussions within the scientific community. According to the authors, in the dimension of context acquisition, the biggest challenge is capturing the user's related context activity. Verbert et al. states that some promising examples of systems, which fulfil to a certain degree the aforementioned objective are the systems that rely on task modelling. In their further argument, the authors underline the emerging need of interdisciplinary collaborative research, which aims to include cognitive science and computer science communities.

Furthermore, in another example in relation to context-aware recommender systems, the authors point out that there are still little research efforts, dedicated to the interdisciplinary aspect of tackling the context acquisition issues [42]. According to the authors, due to the user's predominance within a context setting, it is beneficial to incorporate the cognitive science expertise in order to be able to deal with various factors of human cognitive processes.

To summarise the current chapter and underlying aspects of earlier discussed context-aware paradigm, we present below our understanding, interpretation and use of context-awareness within the proactive computing dimension. Correspondingly, we find it convenient to describe our interpretation of the notion *Context-awareness* by separating the integral elements, and analysing each part individually. The first element of the term, *Context* is characterised in our study as the sum of situational factors, such as general types of a user's online activity, diversity of academic tasks, types of organisational problems, variety of tasks priorities, users' cognitive states, online statistical data and so on. All mentioned elements, therefore, in their various combinations constitute and form a context setting, which is ultimately subjected by our proactive system for its further evaluation.

The second element of the term, *Awareness* is understood as a system capability, allowing to distinguish the pre-defined context elements in their variational sets, and thus to identify a target context situation. In combination, the two elements form the fundamental concept of context-awareness, which is used within the framework of our proactive system. The context-aware paradigm is therefore employed to identify and to evaluate the different factors of a surrounding context, and to analyse a situation with respect to its potential future evolution.

Throughout the various examples of the present chapter, together with our definitions of certain terms, we may consequently notice an emerging need for an interdisciplinary approach as the main prerequisite to guarantee an objective research methodology. Aforesaid condition is essential in order to be able to effectively translate and to implement the different aspects of human behaviour within the system architecture. The given disposition requires an additional knowledge of cognitive-related factors of human mental processes. In the following section we present the main principles and approaches, applied in the objective of providing our study with the necessary cognitive science expertise.

2.2 Cognitive science perspective

In the given section we highlight the interdisciplinary aspect of the current work, which is mainly represented by the theoretical input of cognitive science expertise. In the upcoming paragraphs, we try to view the aforementioned computer science theoretical orientations in conjunction with the corresponding perspectives and methodologies, issued form the cognitive science domain. The given approach allows us to better understand the forthcoming methodological techniques from the twofold perspective of computer and cognitive science research.

2.2.1 Interdisciplinary approach

In its original definition, given by Tennenhouse, for a proactive system to be able to function as an anticipatory and user-guiding mechanism it has to take into account the context, its actors, and their interrelations. Therefore, in order to accurately model proactive scenarios and system behaviour, we have to precisely understand all aspects of a context situation from different perspectives. Only in the given manner, we can endow our system with the genuine context-aware capabilities, by giving it the details about all essential and necessary attributes of a context. Such disposition implies that an interdisciplinary approach is indeed a vital precondition, or characteristic of the entire methodology. In order to build an objective methodological background, we need to direct our attention to different theoretical approaches.

In our opinion, a global shift towards interdisciplinary research model, has been stipulated by the evolution of proactive computing itself. In other words, from the moment of its conception and throughout the time of its evolution, an ever increasing rate of various research attempts has been done for applying the approach in a variety of different domains. In the following paragraphs we try to highlight this aspect by giving several illustrative examples.

The first example demonstrates the use of proactive computing within the framework of event-driven systems. Engel et al. proposes a conceptual architecture for a *proactive event-driven computing* with a perspective of applying proactivity either for eliminating undesired future events, or for taking a benefit of future events, which are considered as advantages [7, 43]. The authors base their concept of the proactive event-driven system on several requirements, depending on the type of environment. In their generic perspective, the given requirements could be characterised as an accurate event-processing methodology and probabilistic predictive and decision making capabilities. Therefore, the authors define the main goal of a proactive event-driven system as reaction to an event that is expected to occur somewhere in the future. The proposed concept of the aforementioned system is then built on a model, which consists of several steps, that is pattern detection, forecasting, deciding, and taking actions.

Another example illustrates the benefit of using the proactive approach within the framework of recommender systems. In their work, Bedi et al. presents a strategy for a proactive recommender system, which aims to accurately assess a context situation and to proactively provide the right suggestions [44]. The authors underline an extreme effectiveness of the aforementioned approach, if applied within a dynamic environment, which is characterised by the constant and rapid change of its state and its attributes. However, the authors equally highlight situations where the effectiveness of the approach decreases, either due to wrongly assessed recommendations, or due to wrongly assessed context settings. Consequently, we may notice a twofold character of the proactive approach.

Another example, which underlines the advantage of *proactiveness* within the framework of recommender systems, seems to agree with the aforementioned opinion of using the approach in a dynamic environment. In order to address an issue of coverage and diversity in recommendations within the domain of news, the authors successfully attempt to employ proactive computing strategies to manage the dynamism of a given environment [45].

Throughout time, the recommender systems have been applied in a variety of domains, which eventually proved to be an effective method of assisting a user. The e-learning environment is therefore, an important domain for implementing and testing context-aware and proactive principles. Thus, Gallego et al. highlights the importance of incorporating proactive recommending behaviour into an e-learning system [46]. The main objective of the highlighted study is to proactively manage the learning material within the electronic platform, according to the needs and interests of a user. The authors build their system basing on the domain-dependent context modelling. The proactive recommendations are therefore based on the analysis of user activity and some attributes of context-related resources' management [47].

Another notable example of employing proactive computing principles is found in the e-health environment. In order to provide personalised healthcare services, context-aware and proactive paradigms are employed for monitoring various context properties, related to the user's activity and his/her health states [48]. The given approach provides a possibility to resolve the problem of healthcare personalisation through the patient's monitoring in a variety of different contexts, such as home, hospital, or any other associated environment. In other words, the aforementioned approach allows the corresponding system to provide healthcare services at the right time and at the right place [49]. Therefore, by applying a proactive biomedical health monitoring, it becomes possible to provide the anticipatory health services, and if needed, to manage the preparation of a medical treatment ahead [50].

The last example that we highlight in the current section, illustrates the advantage of using proactive computing paradigm within the domain of traffic control systems. In their paper, the authors present an effective proactive-based solution for managing the traffic congestion in urban areas [51]. The researchers use a forecasting algorithm, deployed within the traffic control system, which is built on two properties. The first property is characterised by a time interval variable, allowing the system to identify decision thresholds. The second property, on the other hand, takes into account an anticipatory principle of proactive computing, which is built on top of the "reactive" framework of a traffic control system. This example demonstrates furthermore the prospect of interrelations or transitions of possible methodological states, such as from static to dynamic, from interactive to proactive, or from deterministic to probabilistic.

Concluding remarks. The current state of proactive computing research is still considered to be in its prime stage of development [2]. The given disposition allows and inspires us to explore the proactive computing paradigm in different directions. Therefore, throughout the current study we investigate proactive computing and its underlying functional aspects from deterministic and probabilistic perspectives. On the one hand, the deterministic approach allows us to implement the preprogrammed set of context-aware capable rules, in order to enhance a system with the proactive type of behaviour [3, 52]. The deterministic approach, however, represents only one segment of the theoretical framework, applied in the development of the proactive system. In order to enhance the initial deterministic framework with unpremeditated and automated functioning, we apply the probabilistic approach upon the deterministic structure. Therefore, the given methodological orientation allows us to validate the conceptual framework of our proactive system within two theoretical dimensions.

In order to accomplish our initial research objective, which aims to validate the prospective concept of the proactive system, we need to take into account all underlying functional aspects of the system together with context-based attributes of a target environment. As illustrated in the aforementioned examples, proactive computing touches different sides of a context, including the human factor. Thus, it is important not only to take into consideration the computer science research, but also to refer to different factors of cognitive science expertise. In the given perspective, we consider the combination of different study domains to be beneficial, due to its interrelated and complementary effect. Therefore, in order to effectively implement the context attributes, related to different sides of a user's behaviour, we are going to present in the next section an approach, which helps us to model, simulate and to successively translate the various cognitive states of a user into an algorithmic level of system functions.

2.2.2 Cognitive modelling methodology

As stated in the previous section, one of the motives for implementing an interdisciplinary approach, is due to the presence and manifestation of multitude of factors, included into the design of individual proactive scenarios. As we will see in the upcoming chapters, proactive scenarios form an integrated proactive system behaviour. Therefore, certain potential situations are modelled through various proactive scenarios as an abstraction and manipulation of integral context elements. Consequently, the given approach implies the necessity to analyse the user's behaviour in various potential situations with respect to the different aspects of a user's cognitive states. The given analysis mainly consists of an abstraction of context-based user's cognitive and behavioural manifestations.

In order to perform this task, we have to relate our ideas to the theoretical background, which deals with the cognitive side of human behaviour. Therefore, different cognitive and associated methodologies have to be taken into account for a corresponding analysis of users' hidden mental characteristics. By approaching the issue of users' behavioural modelling through the given strategy, we try to define the main guidelines for the design of a proactive system behaviour from the cognitive science perspective.

The design itself consists of specifying certain instances of a user's behaviour, defining its distinct characteristics, and finding the best way of combining and implementing these attributes of user's cognitive manifestations within different proactive scenarios. Given that a user is the chief attribute of a context, we decide to employ the *Cognitive modelling* approach as the main technique for simulating and modelling the different aspects of a user's behaviour. Additionally, as we will see later, this approach allows us to incorporate the statistical evaluation of users' cognitive states into the proactive system. In our opinion, the chosen approach takes into account and agrees with our initial conceptual and methodological goals, allowing to accurately reflect the underlying aspects of human-computer interactions.

The principles of the *Cognitive modelling* methodology emerge as an approach from a more generic theory of *Cognitive architecture*, whose objective is to simulate and represent the structures of human cognitive processes on a computational level [53]. The field of cognitive architecture, in itself represents collateral research efforts, attributed to the domain of Cognitive Psychology, which conceptualises a human being as an information processor [54]. Cognitive psychology deals with a variety of phenomena including perception, memory, attention, reasoning, learning, problem solving, decision making, concept formation and so on. However, despite its orientational diversity, cognitive psychology is consolidated under the common perspective of an information-processing approach, a dominant theoretical orientation throughout decades, providing an efficient way to study human cognition [55]. Correspondingly, cognitive psychology includes four major theoretical directions:

- *Experimental cognitive psychology*, which uses experimental methods for studying human behaviour without involvement of computational modelling.
- *Cognitive science*, which studies the functional aspects of the mind and its processes by using computational models for understanding human cognition.

- *Cognitive neuropsychology* studies the structures and functions of the brain with emphasis on cognitive impairments.
- *Cognitive neuroscience* studies the brain functioning with emphasis on biological substrates of mental processes.

The cognitive science uses computational models in order to understand human cognition, to provide an explanatory basis of a theory, and to help to predict the user's behavioural patterns in new situations [55, p. 4], [56]. Therefore, a cognitive modelling approach within the cognitive science domain is mainly used for representing a cognitive theory assumption for its detailed investigation. Moreover, the given approach is used with the objective to observe how evidence fits the theory. The designated cognitive models are usually designed and expressed as the computer programs, which simulate on the algorithmic level the domain-specific functions of human cognition. The use of the computing cognitive modelling approach, is therefore characterised as beneficial and constructive for generating the specific cognitive phenomena under investigation. Such premise is due to the cognitive modelling capabilities of generating a data, which represents the same characteristics as an empirical data, issued from real experiments, and thus providing researchers with valuable theory-building techniques [57].

The cognitive architecture is the theoretical structure, which represents a generic framework, and provides the possibility for detailed modelling of various cognitive phenomena. The computational cognitive models are then the domain-specific structures and mechanisms of human cognitive processes. Initially, such models are used for theoretical analysis of cognition [58].

In their work, Emond et al. stipulates the beneficial factor of using cognitive modelling within the perspective of human-computer interactions, as the given approach allows the researchers to design and evaluate the methodological techniques and computational models of users' behavioural aspects [53]. The given disposition, according to the authors, consequently provides a possibility to use the computational models within the interactive technologies as the simulated instances of users' cognitive phenomena. Several research efforts have been made to create the aforementioned cognitive architectures in order to simulate the different aspects of human cognition.

Examples of such frameworks are the SOAR system, which was developed by Allen Newell and his colleagues in 1990 with the focus on problem solving and learning, or the EPIC system, developed by David E. Kieras and David E. Meyer with a specific focus on multimodal and multi-task performance [59–61]. The Executive Process-Interactive Control (EPIC) architecture incorporates a variety of structures, which reflect the human cognitive performance as the simulated computer models.

Another notable example is the ACT-R system, developed by John R. Anderson and his colleagues in Carnegie Mellon University [62, 63]. The *Adaptive Control of Thought-Rational* (ACT-R) is a cognitive architecture system, consisting of a set of domain-specific modules, such as a perceptual-motor module, visual module, goal module, memory module and so on. Together, the modules form a network of interconnected micro-systems, which operate and process the information in a parallel and serial manner. The ACT-R system represents the architecture of the integrated domain-specific modules, where it aims to build the scientific evidence on organisational structures and processing mechanism of the brain.

In his work, Durin et al. presents the theoretical background, which serves for creating the technology capable of providing the *"intelligent human-computer interactions"* [64]. The authors use a cognitive modelling methodology for modelling users' cognitive, perceptual, and motor factors for their integration into the user interface. The given approach allows the system to use the cognitive models for evaluating the user's nonverbal information in order to adapt the elements of user interface, if needed, to the characteristics of users' current mental variations. Therefore, the authors' main motives for models' integration is to provide the system with the capabilities of monitoring the user's cognitive states such as tiredness, confusion, frustration or distraction. The authors consider the instances of nonverbal information as indicators of inner mental processes, which are hidden behind behavioural manifestations. Thus, a nonverbal information, according to the authors represents a communication medium for the behaviour interpretation.

In another work, related to the study of human-computer interactions, the researchers emphasise similar advantages of using the cognitive modelling approach. The authors argue that the computer simulations of human behavioural patterns represent an important methodological component for building an effective interactive technology [53].

Within the domain of cognitive science research, there have been many attempts to build the representations of human cognitive processes, using a probabilistic modelling approach [65]. However, a cognitive model, which represents a cognitive function or a state, is essentially the deterministic model, where the main principles of cognitive psychology are mainly based on stochastic representations, and often described by means of probabilistic techniques. Consequently, the given disposition poses a fundamental problem in the relationships between the cognitive modelling approach and the principles of cognitive psychology. A solution, however can be found in developing the cognitive models with the emphasis on probabilistic interrelations between model and data [57]. In other words, the statistical inference can be used in a process of relating the models of cognitive aspects to the empirical data, reflecting the human behaviour [66].

Correspondingly, the approach of using statistical inference to evaluate a cognitive model against an empirical user's data, represents the primary methodological technique of our study, allowing to implement the probabilistic aspects into a proactive computing paradigm. Consequently, the cognitive psychology and cognitive modelling methodology allow us to define and to ground the aspects of human cognition together with the instances of users' behavioural patterns. We use the cognitive modelling approach in order to represent the user-based contextual data on the algorithmic level, and thus to relate the aspects of users' cognitive states to the specifics of the proactive system.

Part II

Methods, concepts and approaches

Chapter 3

Deterministic methodological framework

In the present chapter we define all notions and concepts used in the design and development of a proactive system prototype. The main objective of the following sections is to situate the methodological implementations within the previously defined theoretical framework. This technique will help us to scientifically rationalise all related aspects and characteristics of a system's conceptual framework with respect to the mainstream theories and approaches.

In the following sections we proceed in accordance with our aforementioned research questions (see chapter 1). If the previous section has been characterised and built according to the properties of Research Question #1, then the Research Questions #2 will respectively define the structure of the present chapter. The following sections are divided in two categories. First, we define the nature of *Proactive Scenarios*, which represent the context-aware mechanism, or in other words the perception centre of the proactive system. Here, we describe the role of proactive scenarios, their mechanism of operation and the main guidelines, used in the design and conception of such scenarios. Second, we address the Research Question #2, where we highlight all aspects of the deterministic approach, used in designing the fundamental part of the proactive system.

In section 3.2, we present the deterministic dimension of the proactive system, which has been defined and implemented in the framework of Learning Management Systems (LMS). The main objective of the given section is to delineate the fundamental concept of the deterministic dimension and to subsequently validate it through testing and experimentations. We will describe here the methodology of our first experiment, which aims to validate the deterministic side of the concept through an empirical study, based on enhancements of basic LMS functionalities with proactive context-aware capabilities.

3.1 Context-aware mechanism of the proactive system

In order to better understand the conceptual framework of a proactive system, we will consider its key principles through the perspective of its environment. A proactive system is a goal-oriented mechanism, which aims to bring the feature of proactivity into service-based systems for an enhanced and more adapted operation within its environment. In the given work, we investigate the concept of proactivity as an added-value feature, built in form of a plugin with an idea of its modular integration into a target-system in order to augment its functionality with anticipatory and predictive features. The environment of a proactive system is then thoroughly defined and characterised by an environment of its target-system. Furthermore, the given approach implies that certain changing factors, or context variables of a proactive system environment are partially defined by the fluctuations of the target-system states, which are caused by a user's activity or non-activity on a system.

Additionally, in some cases the variation of target-system states can be caused by its interaction with another system, or by newly acquired data, captured through physical sensors. Consequently, the main functions' characteristics of the proactive system depend entirely on the type of a target-system. In other words, the environment of a target-system defines a type of a proactive system behaviour and its associated proactive services.

As we mentioned earlier in chapter 2, in order to provide a target-system with more intuitive and effective service-based functionality, the system has to be augmented with operational principles of the proactive computing paradigm. The assumed proactive capabilities, have therefore to be integrated through a mechanism, operating on the basis of a concept, which is capable to provide anticipatory and predictive features.

The term *proactivity*, applied within the framework of a target-system, presumes that depending on the currently prevailing contextual situation, the designated system has to do something first or, in other words, it has to take initiative of action. As defined previously, the notion of *proactivity* and its respective concept specify that a primary objective of any proactive system is to initiate a change of a situation, rather than to simply react to an already occurred event. Therefore, in order to act proactively, the system has to be built on the principles of high-level dynamism.

The given approach presumes that for an effective event anticipation and future context inference, the proactive system has to be constantly aware about the current state of context conditions and their characteristics. Consequently, an important requirement for a realisation of the aforementioned functional aspect is the system's ability to interact with the context environment through monitoring, analysis and, if needed, through manipulation of a context data.

In a review of the given account, we can notice one more time that the main prerequisite for an efficient functioning of the proactive system are the context-aware capabilities, which are deployed within a target-system environment. Therefore, in order to operate effectively, the proactive system has to possess a mechanism, allowing to capture the ever evolving states and conditions of its corresponding context environment.

It is important to mention that for the purposes of the given proof of concept, we don't implement and we don't use in our proactive system any physical sensors or actuators. Instead, we employ various techniques and approaches of an event abstraction through algorithmic modelling of context's variational instances. The given disposition provides us with a bigger range of possibilities, allowing to detect different situational aspects. One of such acquired functional qualities is the system's monitoring capability, allowing to capture the finer context details, related to a user's cognitive activity.

Due to the collective and aggregated nature of the proactive system environment, which is defined by different characteristics of the target-system' attributes, including system's internal processes and its states variations, we designate therefore the proactive system's operating domain as a *virtual environment*. As we will see in the upcoming sections, the various aspects and attributes of a context are mainly represented indirectly through different compound elements of a target-system. The context representations are essentially realised by means of allocated statistical combinations of various data instances. In our opinion, it seems therefore necessary and reasonable to appoint the *virtual* aspect to the definition of a proactive system's environment.

Correspondingly, due to their virtual representations, the associated context attributes have to be captured by other means than the physical sensors. In consequence, we focus on developing the mechanisms, which are capable to detect the finer users' context information, manifested within the proactive systems' virtual environment.

The aforementioned mechanism has to be capable to capture the variations of the target-system states, which implicitly reflect the user's activity, manifested within the designated context environment. In order to implement and to generate anticipatory proactive features, we have to provide our system with the possibility to capture the target-system's states fluctuations through the continuous monitoring of its internal processes. Subsequently, in order to realise the given objective, we elaborate the mechanism, which allows the proactive system to monitor the database of a target-system for any state fluctuations. Every data change in the target-system's database consequently represents the user's interactions with the system.

The fluctuations of the database states, will allow the algorithms of the proactive system to interpret and to translate the context details of the user's actions into the computational level of system's functions. As a precondition, such a mechanism requires a thorough prior analysis of a target-system's database, its structure and relationships of its tables. An example of the given technique will be discussed in details in the upcoming section of deterministic approach (see section 3.2).

Alternatively, the monitoring of a target-system can be achieved without a database analysis. In this case, the proactive system performs the direct monitoring of a target-system's user interface for capturing and analysing the user's browsing activity. The aforementioned mechanism necessitates profound prior data specifications with the subsequent arrangement of data into logical and coherent combinations, patterns and ultimately models of the user's browsing activity. An example of this technique will be discussed in section of probabilistic approach (see section 4.1).

3.1.1 Context scenario definition

According to the aforementioned characteristics of a proactive system environment, any change or variation of a target-system state represents an external event or a set of events. On the other hand, a set or sequence of events, associated with a particular context, collectively represent a context scenario, which is defined as domain-specific. Furthermore, the given definition designates that a context scenario represents a particular context situation, consisting of several inherent events, which progressively unfold within the particular time frame.

Considered as one of the objectives of our research, we therefore focus on identifying, specifying and modelling the aforementioned context scenarios. Consequently, in order for a system to be able to monitor the designated context situation with a proactive type of behaviour, specific proactive actions are allocated and assigned to individual context scenarios. During the modelling process of context scenarios, several factors or preconditions are taken into consideration. Thus, given a type of a proactive system environment, the context scenarios are defined as the chain of system events, representing the instances of a user's activity on a target-system. The events are characterised either explicitly in form of human-computer interactions (denoted as physical domain), or indirectly as the corresponding metaphysical representations (denoted as abstract domain).

It should be noted that each element of a physical domain has always its abstract representation or meaning on a metaphysical level. In other words, each context activity of a user provides a set of corresponding cognitive reactions where each reaction is a potential precondition for the next context actions. Any conscious context action is therefore the product of the prior cognitive activity, where the pattern of user's actions is a chain of constant interactions and interrelations between physical and abstract levels. Ultimately, while modelling a context scenario both, physical and abstract domains are taken into consideration with the conditional assumption that the user's behavioural patterns within the physical domain are the aftereffects of his/her inner mental processes from cognitive level.

In our study we understand all context-associated instances of a user's behaviour as outward manifestations of his/her inner cognitive processes and their attributes, such as perception, memory, attention, decision making and so on. The cognitive attributes in this case are characterised by the conditions of a related context. Consequently, our objective is to design a mechanism, which will allow the system to capture the characteristics of a user's activity and its corresponding cognitive preconditions and posterior effects. Therefore, the schematic abstraction of various context scenarios is necessary.

In order to achieve the aforementioned goal, first, we have to specify all relevant and interesting context settings. Subsequently, we have to create the abstraction models of corresponding context scenarios and to integrate them into the proactive system on an algorithmic level in form of computational models. In order to provide a system with the context-capturing techniques for an ultimate proactive mediation, each computational model of a context scenario has the objective to detect its predefined context situation. Every computational model is therefore designed with respect to its abstract context preconditions and the corresponding posterior effects. All computational models, which are the abstractions and representations of a context situation on an algorithmic level, are designated in our study as *Proactive scenarios*.

3.1.2 Proactive scenario definition

Similarly to a context scenario, which consists of several integral events, a proactive scenario consists of several rules or sub-events, positioned sequentially in a logical structure, representing collectively a complete event. A proactive scenario is defined as a set of compound rules, where each rule is an integral action-specific component, involved in detecting a particular, appointed data type. Occasionally, depending on a proactive scenario type, the inherent rules may additionally perform other work than detecting a data. For instance, a rule may launch another rule or a process in order to continue executing the logic of a proactive scenario. A rule is responsible only for a particular, single action, which is performed within a general framework of the confining proactive scenario.

In the given manner, each rule sequentially or randomly aims to perform its unique task, and thus to complete the workflow of a proactive scenario. In other words, the multilayer structure of a proactive scenario constitutes a policy of actions, needed for achieving a context-based task. As in the real world, where each task necessitates the execution of specific actions, a proactive scenario similarly necessitates a policy of actions, which have to be executed for achieving a specific objective of the proactive system.

Hence, each proactive scenario has its own objective of actions, where each objective is defined and characterised by a type of a problem or issue that has to be addressed within a particular context scenario. On the low level, some of the proactive scenarios may have an objective to perform certain internal computations, needed by other scenarios to perform their context-related job.

In a general perspective, the common objective of the proactive system, which is composed of proactive scenarios is to detect and to extract an event from a context situation, to analyse the event, to define the needed response-actions, and to provide the appropriate proactive services, whenever response-actions are needed. However, in order to achieve the aforementioned objective, the inclusive tasks have to be assigned and proportionally allocated between all proactive system modules or, in other words, proactive scenarios. Every proactive scenario, has therefore to perform individually or in conjunction with other modules its unique task within the global scheme of proactive system objectives.

As we may notice, the structure of the proactive system is built on the principles of a modular design. The give conceptual characteristic allows us to add without any system modifications new functional aspects, that is to enhance and to elaborate the behaviour, features and capabilities of the proactive system by simply designing and integrating the new modules (proactive scenarios). The application of the aforementioned approach, consequently allows the system to extend its functional horizons by addressing the deeper and finer details of a context. Hence, a proactive scenario is a single, individual proactive module, built around a particular context situation with its unique action's objectives. All composite proactive scenario's objectives are defined by the needs of a target-system environment.

However, the main objective, common to all proactive scenarios is to act with consideration of future situation development. In other words, every proactive scenario is built to operate with respect to the future, by projecting its actions onto the currently prevailing context setting, and thus deliberately and purposely reshaping its prospective outcomes.

The given condition is a unique and distinctive characteristic of a proactive behaviour, always to act according to the desired future development of a context situation.

As mentioned earlier, if a context scenario is a representation of a complete event, we therefore have to detect its all constituent sub-events first. Due to their modelled structure in form of a rule, the scenario's sub-events are organised and positioned sequentially in a logical structure, which reflects the progressive stages of an unfolding context situation. Upon the detection of one of the sub-events, the structure of a proactive scenario allows an algorithm to know ahead what to expect in the near future, simply by looking on the rest of a model and taking into consideration the sub-events, which have not yet happened. The given system's policy is the main reference, which distinguishes the proactive system from other computing paradigms. Consequently, due to its modular design, where every proactive scenario has the same objective to act with respect to the future, the proactive system may be characterised as an anticipatory mechanism, providing intuitive and proactive interactions with a human agent and other context attributes within the specified environment.

Below we give several approach illustrations of proactive scenarios and their functionality, which aim to demonstrate the anticipatory and proactive features of a system in various context conditions.

Illustration #1 The first example illustrates the proactive system behaviour, designed to monitor and to optimise the satellite's orbit trajectory. A proactive scenario, which is involved in satellite's monitoring, consists of several rules, representing in their logical organisation the sub-events of a potential satellite's trajectory deviation. Namely, each rule has to detect progressively the following context characteristics: (a) the satellite's current position, (b) the satellite's deviation from established trajectory, (c) the satellite's exact position at the moment of the trajectory deviation, and (d) the state of its systems at this moment. Thus, when such an event occurs, a proactive scenario detects and records through step (c) the satellite's exact position at the moment of a problem, and through step (d) it detects that at the moment of the trajectory deviation, the satellite's solar panels have not been displayed optimally in relation to the sun's position. Subsequently, the system sends all related information to a mission control centre for supervision. Consequently, the proactive scenario updates its function parameters by adding recently detected data.

The given approach will allow the proactive scenario to detect on the next orbital turn, through step (a) that a satellite is approaching again the coordinates captured by step (c). Therefore, in order to prevent a problem detected by step (d) in the previous orbital turn, the proactive scenario optimises ahead the position of the satellite's solar panels before it enters the problematic zone.

We can see through the given illustration that already at the step (a), the algorithm may infer the potential future context states by looking on the rest of a model, that are steps (b), (c) and (d), and thus to take all necessary measures before the actual event happens.

Illustration #2 The next example illustrates the proactive system's behaviour, designed for the health state monitoring systems. The given example differs from the previous one in terms of its policy structure, which comprises the capacity of a task delegation. The designated feature allows the initial proactive scenario to launch another service or another proactive scenario in order to take part in building anticipatory and preventive measures. Denoted in the present example as proactive scenario #1, the module is involved in monitoring the patient's health states. The given proactive scenario is programmed to look for any minor, non-threatening health anomalies, indicating first preconditions of a health risk.

As we demonstrated in the previous example, similarly, the current proactive scenario consists of several rules, representing respectively in their logical organisation the subevents of potential health risks. Namely, each rule has to perform the following context-based actions: (a) monitoring a patient for specific minor health anomalies, (b) detecting the specific instances of health anomalies, (c) recording all relevant medical data corresponding to a moment of anomaly occurrence, such as blood pressure, sugar level, bit rate etc., (d) ultimately, upon the condition if step (b) is true, the current proactive scenarios will delegate the task on a higher level by triggering the proactive scenario #2. The second proactive scenario will start to define all necessary measures to prevent the actual health risk and further health aggravation. In due time, the proactive scenario #2 will communicate all needed preventive medical measures to a doctor before the actual health aggravation happens.

Correspondingly, when a health anomaly occurs, the proactive scenario detects and records through step (c) all relevant data, corresponding to the moment of a health anomaly occurrence. Subsequently, the algorithm updates its functions' parameters by a newly acquired data. In other words, the proactive scenario allows the system, through the discovered new data, to optimise its functions. Furthermore, the potential variations of parameters can be added by a doctor beforehand. After providing the functions with the new parameters, the proactive scenario will start to look for a new data upon its next run. Hence, during the next iterations, if the proactive scenario #1 detects the close relationship between patient's current health state and system's new parameters, indicating the values of previously detected patient's health anomaly, it then launches the proactive scenario #2, which will provide this time recommendations to a patient and doctor, before even the minor health anomaly occurs.

In the given illustration, we may notice a slight difference in relation to the first example, in terms of supplemental proactive services, which are involved in building the anticipatory function. The implemented proactive scenarios have to posses the function of a task delegation, which allows the proactive system to allocate the complex context events within several proactive scenarios. Consequently, the initial proactive scenario will have to activate and to accordingly parametrise the next proactive scenario, whose objective is to carry on the initial task on a different level of the context situation. As we have seen in the example above, both proactive scenarios in the cooperative manner are able to provide the doctor with the possibility to acquire the vital data proactively, before the actual health risk occurs.

Illustration #3 In our last example, we present a new type of proactive scenario, which incorporates the functions of anticipation and prediction through a slightly different mechanism. The scenario characterises a situation, where the future development of a context event consists of several possibilities. An example is represented by a situation, where a user may manifest either a state of mental satisfaction or dissatisfaction, in relation to the obtained results during a web search querying. In order to estimate an occurrence of the most prevalent context variation, the probabilistic calculations may be applied upon the patterns of a user's activity. In other words, probabilistic calculations are applied in order to evaluate the user's cognitive states of satisfaction or dissatisfaction, expressed during an online search activity. In order to achieve the aforementioned objectives, the cognitive models of user's mental variations have to be created. Consequently, upon the positive probabilistic evaluation of one of the cognitive models, certain measures can be undertaken in order to provide the user with more relevant results. The given example will be described and discussed in more details in section 4.1.

The presented illustrations of proactive computing approach demonstrate several design possibilities, allowing to implement the feature of context state's inference. Every example illustrates its own technique. In the first example we demonstrated that an event anticipation can be achieved through monitoring the logic of a proactive scenario, where the algorithm takes into consideration the ratio of happened and non-happened sub-events. In other words, the proactive system is able to use the situation development in itself as its input data, by using proactive scenarios for correlating the levels of their accomplishment with the particular states of a context situation.

The second example illustrates the technique, where an event anticipation can be achieved on a higher level, through the currently active proactive scenario. In the given approach, a proactive scenario may launch another proactive scenario, which will carry on the initial objectives, pertinent to the same context event, by initiating the proactive actions with respect to the desired future outcomes. The designated mechanism is a representation of the general algorithmic logic of our proactive system, which allows its functions to select and to activate the next most relevant and needed proactive service in order to deal with a situation at hand.

With the ability to use the real life scenarios of an evolving context as its input sources, the proactive system is therefore capable to project its target actions with respect to the potential future development of a given context situation. Moreover, in our last illustration we demonstrated that an event prediction or evaluation can be achieved through the application of the probabilistic inference. The given approach is applicable upon the condition, if the target context characteristics presume several possibilities for the context variations. The system then estimates the most probable context deviation. It should be noted that context-interpretative algorithms are not limited to only one anticipatory technique. The application of several approaches and their combinations may be used in the design of dedicated proactive scenarios.

Furthermore, the aforementioned examples demonstrate the underlying principles of the proactive computing, mentioned by Tennenhouse [1, 2]. Notably, due to their high frequency operation, most of the system's computations and data exchange are done without human interference into the process. Nonetheless, as we saw in our examples, the proactive system always remains under the supervision of a human agent.

The presented approach illustrations demonstrate an important methodological characteristic, which stipulates that a proactive system's behaviour has always to be oriented towards a particular target environment and its associated context variations. Only in the given manner, an assortment of distinct proactive scenarios can be built in order to accurately cope with the context-specific events. Therefore, we design and build our proactive scenarios, representing a general system's behaviour, by focusing on the real life context situations. The particular context events are then carefully elaborated with respect to the proactive system's objectives, that is to operate with regard to the needs and goals of an external agent. On the design level, every context event is modelled to accurately reflect the aspect of a context transformation from one state to another. The given approach stipulates that every context event is either a precondition or a product of another event.

By employing the aforementioned approach, we are therefore able to design in a seamless and a logical manner the general flow of a situation development. The proactive scenarios may in general have different areas of application within a context category. The given characteristic typifies the nature and the main operational directions of a scenario. Moreover, according to a variety of possible context attributes and a diversity of issues that may arise from a context setting, the scenarios may differ considerably in their features and complexity. Some of the proactive scenarios may have a very simple logical structure, where others can be characterised by distinctively complex organisational aspects. In the latter case, the implementation of additional computational techniques, such as probabilistic data estimation is sometimes required.

Consequently, a system, which is enhanced by the proactive behaviour, may use various proactive scenarios in order to deal with problematic context events by simply employing the appropriate event-oriented target actions. Every proactive action is therefore stipulated by the inherent objective to resolve the emerging event-based issues. As we saw in our conceptual description, the proactive scenarios are used as the main technique for enhancing the target-system's capabilities with the proactive features. Successively, the given approach allows the target-system to deal more effectively with the context events of an associated environment.

The highlighted above conceptual characteristics of proactive system's mechanisms demonstrate the main attributes, which allow the proactive system to cope intuitively with an associated context environment. The key prerequisite for designing and building an accurate proactive system behaviour is mainly stipulated by a type of the target system's environment. The given approach disposition allows us to define the character of the proactive behaviour and the underlying mechanisms of its actions. All proactive actions are then designed and implemented in the framework of proactive scenarios, which concurrently serve as context-aware and context-reshaping mechanism of the proactive system.

On a conceptual level, all proactive scenarios are designed and based on the real-life context scenarios, which are carefully selected and further elaborated around a particular context event. The given approach is the fundamental condition for defining the proactive scenarios' main objectives. Due to the high number of possible context events and their respective individual objectives, the structure and complexity of corresponding proactive scenarios may therefore vary considerably. Nonetheless, an objective, common to all proactive scenarios always remains the same, that is to act with respect to the desired future development of a context situation by taking in consideration the needs and objectives of a user or another system.

In the next section we will discuss all conceptual issues, related to the design and implementation of proactive scenarios. We will take a look on the main attributes of proactive scenarios, where we study their relationship to the cognitive science expertise.

3.1.3 Conceptual framework of proactive scenarios

According to the conceptual characteristics and mechanisms of the proactive system behaviour, we know that all proactive scenarios by definition naturally inherent all related aspects of corresponding context scenarios. Distinctly, all attributes of context scenarios become the guiding milestones, which direct the process of designing and developing proactive scenarios. In the given subsection we highlight the most important attributes, which characterise the nature and the principles of proactive scenarios.

Hence, we divide the process of attributes' specification in two categories, the attributes of physical domain, related exclusively to the context settings, and the attributes of abstract domain, related to the cognitive aspects of a user's behaviour. The argument for such category differentiation is due to the governing premise, which defines a user as the main actor of a context. The given postulation, consequently leads to the methodological disposition, which defines the user's behavioural patterns, manifested on a physical domain as a reflection of corresponding cognitive activities, initiated on the abstract level. In other words, every human action, performed in any context situation is a posterior product of various cognitive preconditions, which are formalised either by other cognitive aspects or by circumstances of previously experienced context events. In our opinion, it is therefore important to dedicate a substantial part of our research to investigating equally the mechanisms and attributes of veiled connotations of an abstract domain, represented by the user's cognitive characteristics.

A context scenario, as we already mentioned, is primarily characterised by an environment, where the user's activity is taking place. In our study we investigate the nature of the proactive computing paradigm within two types of environment, that is the academic e-learning platform and web search engine.

The first environment is represented in a form of an e-learning platform, or Learning Management System (LMS), which functions as an online academic tool provided by the University. The main user types are students and teachers. The given disposition presumes that the academic-based interactions within the given environment are the main types of activity. More specifically, the platform interactions between users are outlined, but not limited to such activities as management and organisation of the course-related content, assignment management, bibliography management, tasks organisation, announcements management, course-related forum chatting, blogging, course scheduling and so on. Thus, we may derive that the given environment is mainly characterised by an intermittent and scattered character of interactions between teachers and students. Occasionally, the interactions may involve a communication interchange with the system administrator or study department.

We may notice that the design characteristics for proactive scenarios, operating within the given environment, must have very specific context event orientations, such as the guidance of a user in his/her daily academic-related online and offline activities. The user is then defined as a teacher or a student who uses the e-learning platform according to the needs of his/her course-related objectives. A teacher, for instance, may post an assignment and provide the students with all related information by posting the assisting documents or, by sending the corresponding assignment guidelines through a messaging system. On the other hand, a student has to view the assignment posting by referring to all supporting information, provided by a teacher before the task deadline. Ultimately, the student has to accomplish the assignment and to upload it in due time into the electronic platform.

In the provided example, we may notice a context attribute, which is represented by the teacher-student interactions, and is characterised by the governing objectives of these interactions. Basing on the aforementioned example, we are able to derive a general list of attributes, which are common to most context events of the given environment. Thus, by defining a general list of context characteristics, we create a schema, or in other words the guiding principles for designing accordant and adapted proactive scenarios, which will have the potential to accurately deal with the associated context events. Below, we present a general list of attributes, pertinent to an LMS environment.

Attribute A1: user interactions. All context attributes of the given environment are influenced by, and based on the electronic platform interactions, stipulated by the course-related activities of involved parties. The mentioned interactions are characterised by a bidirectional information exchange in form of the teacher-student or student-student communications. The first context attribute, denoted as A1 and symbolising the interactional aspect of the LMS environment represents the ultimate product of all engaged sub-attributes. The given disposition implies that in a process of attributes' identification, we go from evident, distinct elements, such as user's interactions, to the abstract, defining properties. In a general perspective, the latter underlying elements often provide a ground for the overt, distinct elements of a context. Thus, the A1 attribute, representing the interactions between teacher and student, characterises nearly all types of platform-based activities. The interactions may take the form of forum postings, assignment submissions, messaging, blogging and so on.

Attribute A2: subject of interactions. The next context characteristic is the sub-attribute, related to the aforementioned interactional perspective of the LMS environment. The given context attribute is denoted in our study as the sub-attribute A2. More specifically, it represents the subject of the aforementioned teacher-student interactions. The *subject sub-attribute* is often represented as the teacher-provided, course-related task, which has to be accomplished by a student. In the perspective of our example, the A2 sub-attribute (teacher's assignment to students) represents the underlying basis for the attribute A1 (teacher-student interactions). In other words, the A2 sub-attribute provides the ground for all subsequent user's interactions, carried out within the frame of the A1 attribute. The A2 sub-attribute respectively defines and characterises the properties of the A1 attribute, meaning that characteristics of the teacher-provided tasks define the nature and traits of the successive teacher-student interactions.

Attribute A3: characteristics of a subject. The following context attribute, denoted as A3 delineates the characteristics of a subject, or in other words, specifications of the sub-attribute A2. The given A3 sub-attribute aims to define all aspects, which characterise and specify the nature of the subject (A2). The subject's properties can be illustrated by a level of importance, composite complexity, involved time frame, subject's purpose, category and so on. In the perspective of our example, the mentioned properties are the aspects or information, which characterises the assignment task.

Attribute A4: subject-based activities. Correspondingly, if the previous attribute aims to define the subject, the present context attribute has the objective to specify the subject-based activities, which must be carried out in order to accomplish the main task (A2). The present context attribute is denoted in our study as the sub-attribute A4. The given sub-attribute aims to define and to specify all key activities or actions, which are involved in the accomplishment of the course-related task. In relation to our example, in order for a student to be able to accomplish the task at hand, he or she has to perform first a background search and analysis of the relevant literature, to review the course notes, or to perform any other activity, which is necessary in the effective achievement of the given goal.

Attribute A5: characteristics of subject-based activities. The last context attribute is linked to specifications of the subject-based activities. Similarly to the A3, the present sub-attribute, denoted as A5 aims to define and to specify the properties of potentially involved activities of the sub-attribute A4. From the perspective of our example, the present sub-attribute has the objective to provide the actions-related meta-information, allowing to identify and to characterise all necessary and potential activities, which have to be performed by a student in order to accomplish the task at hand.

As highlighted in the list of context attributes, the definitions of context characteristics are stipulated by distinct, evident elements at the beginning, and by more abstract, underlying properties in the end. The gradual shift of the defining attributes' qualities, results in the division and categorisation of their constituents in two methodological classes, physical and abstract domains. Both classes have their distinctive, individual characteristics, which become detectable through the application of various algorithmic techniques. The structural attributes of the physical domain, easily distinguishable and evident at the first sight are characterised by the user's explicit interactions with the system. The given interactions are usually stipulated by various forms of the user's online activities, performed through the graphical user interface. Consequently, the user's actions and the context attributes of the physical domain are easily detectable through different computational techniques. For instance, the context characteristics, representing the traits of user's interactions with the system can be identified through monitoring the user's interface-based activities, including mouse clicks, mouse hovering, strings inputs and so on.

The given orientation allows us to distinguish and to select the most interesting, relevant and consistent types of interface-based activities for their further analysis and sorting into the coherent detectable patterns. Subsequently, the progressive detection of the designed activity-patterns will allow the algorithm to interpret and to evaluate the different aspects of a context setting, and thus to reveal the prevailing event inclinations. The specific context event can in this regard be represented by a thoroughly defined sequence of various interface-based activities, such as combination of mouse clicks, pauses, mouse positions and so on. In order to accurately define all physical patterns of a user's online activity, we need to thoroughly define its abstract implications and meanings. Hence, our main objective in the design of proactive scenarios is to specify the physical context events, displaying the nature of user's settings conditions by applying an abstract analysis of the involved attributes.

If the main objective of a physical domain methodology is to identify the relevant context events, consequently, the main goal of an abstract domain methodology is to identify and to establish their underlying specifications and properties. For these purposes, an identified context event has to be successively translated into an abstract level for its further specification analysis. The principles of the given approach presume that the transition to the abstract domain will eventually involve the context attribute's abstraction and categorisation. Moreover, if a scenario's objective is set to deal with the variations of human behaviour, the transition to an abstract level will additionally include the consideration of the human cognition and its associated processes.

In order to design the accurate relationships between context attributes, we have to analyse them further on the metaphysical level of the properties formalisation. In other words, during a context analysis, we try to create the relationships between cause and effect of an event by studying its preconditions and identifying its most probable posterior effects. As we mentioned earlier, in relation to the human factors, in our study we understand the instances of a user's behaviour as the posterior product of his/her cognitive preconditions, formalised by previous experiences and existing knowledge about the problem. The attributes of an abstract domain are additionally characterised by the user's cognitive processes, involved in a task accomplishment. A situation, where the context events involve human participation, the abstract context characteristics are often represented by cognitive attributes of the user's mental activities.

Concluding remarks. Due to the fact that certain proactive scenarios deal with human factors, involved in a context setting, we therefore need to consider our methodology from the perspective of different cognitive approaches. The given disposition allows us to specify the aspects of a user's behaviour, and thus to better understand its underlying principles and characteristics. Consequently, the aspects of a user's behaviour are expressed through the user's cognitive processes and their associated attributes, which are defined in our study on three levels.

The primary level denotes the *cognitive precondition factors*, which are defined as the user's previous similar experiences, expressed through collected skills and competences. The memories of previous problem encounters, which are correlated with user's competences are mainly expressed through his/her performance during the task accomplishment.

The secondary level is characterised by *existing knowledge*, which is often formed either through the experience accumulation and event observations, or targeted learning. The third level, represents the defining factors of desired situational outcomes in form of the *posterior effects*. The given factor characterises the user's future behaviour, which is based on the current context state and the related existing knowledge. All aforementioned human factors, involve the activity of several essential cognitive prerequisites, such as perception, memory, attention, reasoning, problem solving and decision making.

Consequently, while creating and designing proactive scenario, we have to analyse the user's behaviour in various potential situations and to define its stipulating cognitive aspects. In order to perform the aforementioned task, we have to analyse the user's behaviour with regard to various cognitive theories, dealing with the cognitive characteristics of a human behaviour. However, in order to keep our research topic focused and narrowed down towards the defined objectives, we leave the associated cognitive theories generic, which furthermore allows us to create a space for open discussions.

3.1.4 Summary

As highlighted in the discussion above, a design of the proactive system behaviour inherently includes a variety of factors and attributes, which in a compound form, allow us to build a mechanism, capable of anticipatory and predictive features. The design process conventionally consists of several milestones, used as the guiding principles in modelling the proactive system behaviour. Below, we delineate the general steps of this process in form of a summary, which aims to outline and to recapitulate the main points of the given section.

The central point, discussed in the present section highlights the design principles and implementation mechanisms of proactive system's capabilities for context-awareness. According to its definition, the system's capacity for context-awareness primarily comprises the need of sensors' use. However, due to the distinct, computerised type of environment, used in our study, which is characterised by a target-system environment, we resort to other means of context detection than the use of physical sensors and actuators. Instead, we design and develop a mechanism, represented by proactive scenarios, which is equally capable to detect the details of a context situation by monitoring its continuous evaluation.

All proactive scenarios are designed upon, and based on the real life context scenarios, which are taking place in a particular situational setting. Our design objective is therefore to base the context-aware capabilities of our system on the functions of proactive scenarios, whose goal is to represent an abstraction of a context event through different computational models. One of the key advantages of using proactive scenarios' mechanism is stipulated by their built-in capabilities to always act with regard to the future development of context situations. There are several characteristic steps, which need to be taken into consideration in the design process of proactive scenarios.

The first step, a precursor in a scenario's design, is a definition and specification of a problem that has to be addressed and mediated by the proactive system. In order to accurately define the problem characteristics, we therefore try to specify first the problem structure and its constituents. In other words, in order to create a model of a problem, which may include a multitude of context elements, we try to imagine first what the problem should consist of, that is to define its composition. The elements of the designed model, representing a particular problem, are hence our aforementioned context attributes. Each contextualised model element, pertinent to a physical domain, always has its abstract representation, or corresponding meaning on a metaphysical level.

According to the aforementioned conceptual description, every representation or meaning on a metaphysical level is, in fact, a corresponding attribute of an abstract domain. Moreover, every abstract domain attribute is generally characterised by several defining aspects. Consequently, an attribute of an abstract domain is defined by its precondition factors, existing knowledge about the related problem, and potential posterior states, which often characterised as desired, potential outcomes. It is important to mention that an abstract attribute may be represented either by general properties, relating to simple meanings and representations of a physical domain, or by cognitive-related properties, associated with human mental processes, related to different activities in a physical domain. In the latter case, the element of *precondition factors* of an abstract attribute definition will change to *cognitive precondition factors*.

After having defined all context attributes of both dimensions, the next important step in creating and designing proactive scenarios is the process of accurate and coherent attribute interrelation. In other words, in order to simulate the defined problem, we have to correlate all attributes of a physical domain with the corresponding attributes of an abstract domain into a coherent and logical structure, which will characterise our context event. Depending on the problem definition, both orientations of the attributes' correlation may be used, that is from physical to abstract and/or from abstract to physical. For the coherent attributes' interrelation, we have to decontextualise our event by removing it from the context in order to define its pertinent generic properties.

An example of the aforementioned decontextualisation can be presented in form of a question that we have to formulate in order to define the event's general characteristics. The given questions may have the following form: *What are the general characteristics of a student's low efficiency during the task accomplishment?* or *What are the defining aspects of the user's cognitive states of satisfaction during an online search?* and so on. In other words, we try to define first the meaning of a state or a process, which represents an event outside of its actual context. After having defined the necessary event characteristics during its decontextualisation, we ultimately reassemble the event and analyse it in the frame of its actual context, with respect to its previously defined general characteristics. Therefore, the given process of an event contextualisation is mainly stipulated by the aforementioned attributes' interrelations. During an event contextualisation, we try to match all context attributes of both dimensions by assembling in a logical structure the elements of a user's context activity and the elements of his/her current cognitive states.

The final methodological step in creating proactive scenarios is characterised by the process of event abstraction. In other words, we have to translate our event into an algorithmic level through the logical representation of its abstract domain attributes by detectable computational instances, manifested either through the user interface or the system database. Therefore, the process of proactive scenarios' design and conception mainly consists of an accurate event definition, and its successive abstraction into the computational level.

In the present section we illustrated and characterised the main concept definitions of the proactive system behaviour, which involves a multitude of inherent factors. We investigated the underlying principles of proactive scenarios and their corresponding operational mechanisms on the theoretical level. The given disposition allows us to build the necessary concept framework for further practical system implementations. In the upcoming sections we are going to see the implementation details of the proactive system behaviour, applied within two types of environment, first in a target environment of Learning Management Systems (section 3.2), and second in a target environment of web search engines (section 4.1).

3.2 Deterministic approach: proactive computing in the framework of Learning Management Systems

The main objective of the present section is to address and to investigate the underlying aspects of our Research Question #2, which aims to highlight on an empirical level the deterministic side of the proactive system. In the current section we present the details of a system's deterministic methodology, which has been practically implemented and tested within the framework of Learning Management Systems (LMS). Our objective here is to delineate the characteristics of the fundamental concept of the deterministic approach and to validate it through subsequent experiments. Below, we present the first part of the system's proof of concept, which aims to validate the deterministic side through an empirical study by enhancing the basic LMS functionality with the user-oriented, proactive, context-aware capabilities.

3.2.1 E-Learning environment

Before proceeding to the description of methodological principles, related to the deterministic approach, we first have to characterise the underlying nature of our target-environment. As we mentioned earlier, the main operational environment of the proactive system is entirely defined by a type of the target-system. Hence, the basic methodological principles, characterising the deterministic side of the proactive system are going to be applied and tested in the framework of Learning Management Systems. For the purposes of more adapted and effective application of our approach, we have to thoroughly understand all historical and evolutionary traits, which form the underlying principles of the current e-learning system paradigm. The given disposition is important as it allows us to identify and to characterise the target-system's strong and weak points of its functional principles. Consequently, during a process of the LMS enhancement through an integration of the designed proactive features, the identified system's characteristics will allow us to take advantage of the strong points, and to respectively reinforce the weak points. Ultimately, the discriminative target-system analysis will allow us to design adapted and better fitted proactive behavioural orientations, and therefore more intuitive proactive services.

We attempt to define the key notions of e-learning systems paradigm by taking a brief look on general evolutionary traits of tutoring systems, which have their roots in the popular educational theories and learning methods. During his research on adapting teaching technologies, Bloom et al. has underlined the importance of one-to-one teaching approach by highlighting its effectiveness in a general learning process [67]. In his study, the author showed the results' difference of two teaching approaches, tutoring and instructional, where the students, tutored individually were more successful in their final results with an average outperformance of 98%. Consequently, the results of Bloom's study opened the doors towards many other applicational directions, allowing the given model to be employed in various related fields.

The idea of the computer-assisted instructional programs in the computer science field, has already dominated as one of the main focus areas in many research labs [68]. However, the status of the given computer-mediated learning models still needed a substantial contribution of the additional research. The first significant input has been made by J. R. Carbonell in the early 1970s during his Ph.D, where he developed and adopted the tutoring model of the first intelligent tutoring program SCHOLAR [69, 70]. The objective of the aforementioned human-stipulated, tutoring model was to assist a learner in his/her education-related activity. In the later research, investigating and studying the theories of learning, the importance of feedback, practice and learner guidance has been highlighted and accentuated by many researchers [71–73]. In the field of human-computer interactions, the importance of a tutorial feedback is highlighted by numerous attempts of implementing the computational models of the feedbacks into various electronic educational programs [74].

The Learning Management System, abbreviated as LMS is a product deviation from the mainstream concept of e-learning systems and computer-assisted instructional programs. An LMS platform represents an online environment, oriented towards managing the different sides of blended learning, such as administrative task management, organisation of different aspects of course-related activities, bibliographic material management, messaging, course scheduling and so on. The main goal of the given e-learning platforms is to provide facilitating management solutions for various academic activities to students or teachers who use the LMS environment as the main tool for course-related, educational tasks.

One of the notable examples of the Learning Management Systems is the *Modular Object-Oriented Dynamic Learning Environment*, or simply *Moodle*. The Moodle software has been originally developed by Martin Dougiamas, whose goal was to create a service-based electronic platform in order to help the teachers in managing their online courses [75]. According to the system's documentation, describing the Moodle philosophy, the e-learning platform features comprise several approach orientations, including *Constructivism*, *Constructionism*, and *Social constructivism*.[1]

In summary, all three notions stipulate the idea that a knowledge can be built through several techniques, which are characterised as (a) the active interaction with learners' environment, (b) the course material interpretation and its logical organisation into comprehensible structures, and (c) the collaborative knowledge sharing. The key point of the aforementioned philosophy is to highlight the bidirectional approach in a learning process, which stipulates that besides teacher's input, the learners themselves can as well contribute to a general framework of learning activities. Consequently, the Moodle platform can be used in a variety of active domains, such as the educational and training environment, business, administration and so on. The key features, pertinent to the Moodle functioning can be characterised as follows. The Moodle provides a user with the functions of general management of course-related materials, assignment submission, file management, assignment grading, calendar, news and announcement posting, forum discussions, instant messaging, glossary, quiz, wiki, text editing and so on.[2]

However, in spite of the large list of the aforementioned features, which make the Moodle platform a popular open-source course management system, we must highlight that it lacks certain feature capabilities of more complex computing methodologies. As we saw

[1] Retrieved from: http://docs.moodle.org/27/en/Philosophy
[2] Retrieved from: http://docs.moodle.org/27/en/Features

in the Moodle's features characteristics, given above, the platform provides the user with the fundamental course management functions, which are essentially built on the reactive and interaction-based computing principles. The Moodle operational features do not include (at least for now) more complex functions, such as user's behaviour monitoring, inference of context situations for the content recommendations, proactive notifications and so on. We assume that the original Moodle conception strategy didn't intend to include in its development more complex computing approaches, but had the objective to provide the user only with the necessary course management functions. Given that the platform provides the perfect course managing functions, we intend to use the Moodle platform as the basis for the design, implementation and testing of more complex computing paradigms, such as proactive computing. Correspondingly, due to its open-source type, easy availability, academic and managing orientation, and plugin-supporting functions, we choose Moodle platform as a perfect testing ground for implementing and validating the deterministic part of our proactive system.

3.2.2 Amplification of Moodle's basic functions

As we may notice, the Moodle's fundamental functions and features can be characterised as basic in comparison and relation to more complex computational methodologies. The given target-framework characteristics provide us with the necessary and relatively flexible *action-space*, allowing to implement and to test new computational approaches and techniques. The aforementioned *action-space* represents a vital strategic choice, characterising our methodological objectives of the proactive system implementation. In other words, the given strategy can be defined by our deliberate intention to use the Moodle, representing a stable system framework for the purposes of testing the new, not yet fully developed experimental concept.

It is important to mention that due to the effectively developed methodological strategies, the aforementioned experimentations of proactive system behaviour do not put a user at any kind of associated privacy nor cyber security-related risks. Therefore, the combination of target-system's characteristics and our initial methodological objectives, represent one of the justifying motives for choosing the Moodle platform to test our proactive module. The underlying principles of implemented and tested features have to reflect the deterministic part of a proactive behaviour, and thus to fulfil our first research objective to validate the foundational part of a proactive framework approach. An extension and elaboration of Moodle functions, involving the process of new features design, requires furthermore the applicable implementation techniques and subsequent functionality testing. In other words, the main challenges, stipulating the Moodle enhancement are characterised by such empirical issues as finding the suitable approaches, allowing on the one hand to successfully embed the proactive module, and on the other hand to test its functionality.

In order to address the first issue, we take an advantage of the Moodle's plugin support, which represents a suitable solution for resolving the implementation problem. The defining strategy for the code writing is characterised by, and oriented towards the development of the proactive system as a Moodle plugin. The given approach, highlights an important technical aspect, which defines the integrated proactive system as the framework isolated module. On the other hand, the given methodology implies that we don't change the target-system code, but instead, we simply fuse the proactive system behaviour through a modular technique of plugin integration. The framework isolation, is therefore our main approach directive, characterising and defining the key criteria of the development process. The given theoretical disposition allows us to build the self-sustained, semi-autonomous proactive system, which is programmed to always operate beyond the defining limits of a target-system.

In addition to the aforementioned motivations, characterising the technical side, another motive for using the Moodle in our research is stipulated by its built-in collaborative mechanisms, which imply an important condition that the same system modules can be simultaneously used by several users. The given aspect provides the conditional possibility to design and to test the unique proactive features, aiming to operate within the context environment, which includes the participation of multiple users. The corresponding features of the proactive system are therefore programmed to take care of group objectives and their learning-related collaborative needs.

The aforementioned proactive features have the objective to aptly address the aspects of an online collaborative work, which is successively characterised as a vital prerequisite for collaborative learning. On the other hand, according to the Moodle's inherent features, we know that the e-learning platform takes an individual approach to course-related learning activities. The given disposition provide us with a possibility to focus on the design and development of corresponding proactive features, which equally take an individual approach to the user's activity. The main aspect of our system's operation is conditioned by the proactive system behaviour, which additionally provides the context-intuitive, response-actions. Both, individually-oriented and collaborative-specific proactive features have the common objective to increase the effectiveness rate of student's learning process, which is partially dependent on the use of the e-learning platform as an assisting online tool. Therefore, all instances of added features are based on, and designed according to the defining principles of the proactive computing, which in succession plays the role of the main instrument of the Moodle's enhancement.

From a theoretical point of view, the enhancement of the Moodle platform with various proactive features, will naturally change its original conceptual characteristics and will augment its basic functional orientations. The given conceptual shift will inherently contribute to the upgrade of the Moodle's fundamental functionalities, and will move its methodological framework towards a dimension of more complex computing paradigms. One of the distinctive characteristics of such conceptual shift can be characterised by the system's acquired capabilities of semi-autonomous functioning, which is primarily defined by the system's potential to operate without explicit instructions from a user. Consequently, by referring back to the aforementioned Tennenhouse's premise of proactive computing (see section 2.1.1), such conceptual orientation implies that Moodle can operate on a higher frequency level, without constantly waiting for a human feedback. Furthermore, the given methodological setting allows the Moodle's operational framework to shift from the basic interactive orientation towards a more advanced proactive computing paradigm.

For a learner user, the mentioned disposition commonly implies a constant academic guiding in user's daily learning activities, which is considered as a valuable study practice from an instructional and pedagogical perspective. Moreover, such "tutoring" type of system behaviour may serve the learner as a potential motivating factor in his/her studying process. In the case of an effective realisation of the aforementioned goal, for a teacher, the system's assisting functions may signify the potential and partial tasks discharge, which are commonly related to the supervising and monitoring aspects of the course organisation. Consequently, the enhanced e-learning platform may provide a teacher with the valuable academic e-assistant. As might be expected, in order to fully realise the aforementioned objective, the target e-learning platform has to be enhanced more than by few simple proactive scenarios. All further system design and elaborations, requiring additional research efforts, are considered to be beyond the scope of the current PhD thesis, and are thus ascribed as guiding objectives for future work. However, the experimental realisation of the aforementioned objective may serve as a stable, empirically valid proof of concept and a guiding schema for the future design and development of electronic academic assistants.

The central mechanism, providing the system with a proactive behaviour is based on the design and development of corresponding proactive scenarios, where every scenario is programmed to take care of a specific context setting. Consequently, the same methodological orientations are valid for the enhancement of the Moodle's basic functionality. In order to fulfil the given objective, we have to build the corresponding type of proactive behaviour, basing on the design and development of the Moodle-specific proactive scenarios. Successively, the Moodle-based proactive behaviour will be characterised and entirely conditioned by the type of its proactive scenarios, their complexity and orientation. Our main objective is therefore to design and to develop proactive scenarios, which on the one hand must enhance Moodle's basic functions, and on the other hand to reflect the needs and objectives of user's learning activities.

In order to accomplish the latter goal we have to additionally find the suitable proactive solutions for covering the aspects of a user's activity, which are not supported by Moodle. We can logically assume that the instances of a user's activity may include a multitude of corresponding behavioural patterns, and thus to cause substantial context variations. However, according to the objectives of the current study, we focus on the design and development of proactive scenarios, which only cover the most important part of the user's learning activity. Consequently, the given proactive scenarios will help us to validate and to empirically ground open theoretical concepts and approaches. In the upcoming sections we will give exhaustive details, describing the types and variations of the applied proactive scenarios.

3.2.3 Deterministic attributes of the proactive e-learning system

As we mentioned in our earlier subsections, the main proactive mechanism and its corresponding system behaviour is based on the design and implementation of proactive scenarios, serving concurrently as a means for context-awareness and event-specific targeted mediation. In the given example of e-learning platform, the mechanism of proactive scenarios represents the main principles and foundations of our deterministic framework. In the earlier subsections we have described the conceptual characteristics of proactive scenarios, consequently, in the upcoming subsections we discuss the mechanisms of their functional aspects and practical implementations. We conceptually divide the constructive description of deterministic framework principles in two compound levels.

In order to describe the first-level principles, we have to highlight first the inherent prerequisites and criteria of scenarios' organisation as well as their corresponding classification. We have to distinguish and to group our proactive scenarios into several concept categories, which respectively specify the scenarios' key characteristics. The need for scenarios' categorisation is partially due to the operational logic of the proactive module, which has to be optimised in order to provide an effective, resource-economic functioning. The area of scenarios' application delineates the fundamental structure of a proactive scenario. The scenario's composite functions are exclusively adapted to a particular type of a context event. The scenario's function specification allows the proactive system to be very accurate and precise in relation to its target actions. On the other hand, the operational logic of a proactive module defines the global organisational policy of scenarios' execution. The given directive policy, plays an important role in an effective structuring of the main algorithmic processes of the proactive system.

According to the aforementioned specifications, all proactive scenarios, related to the Moodle environment are classified in two main categories, that is *Meta-scenarios*, abbreviated as MTA, and *Target-scenarios*, abbreviated as TRG. As we will see later, the given classification plays furthermore an important role in relation to the applied methodological strategy, which allows us to optimise the system's triggering mechanisms. In a general perspective, we distinguish two types of scenarios' orientations, Meta-scenarios, which are mostly defined as system-oriented, and Target-scenarios, which are characterised as action-based. More specifically, Meta-scenarios represent a mechanism, whose objective is to monitor all system changes, reflecting the current state of a corresponding context setting. On the other hand, the Target-scenarios are programmed to discharge the context-specific response actions, and thus to provide the targeted mediation of an event, basing on the previously detected context data, which has been captured and supplied by a dedicated Meta-scenario.

In the description above, we may notice the first algorithmic mechanism of task delegation, which works as the main prerequisite for channeling the information flow from one proactive module to another, represented respectively by MTA and TRG scenarios. The mentioned approach plays an important role in building and optimising the proactive system's launching mechanism. The given methodology is necessary for several objectives, including memory optimisation, cost-effective operation, system action specialisation, and most importantly for the generic and modular structuring of proactive system behaviour. Consequently, we may notice in the proactive scenario's structure the first-level, deterministic principles, where two types of "independent" processes are programmed to work in constant interrelation in order to achieve a common goal. It should be noted that on the given deterministic level, the process of a context derivation and targeted action assignment does not include any instances of probabilistic estimation nor any data evaluation. The entire process operates as a result of a task delegation and is fully based on the interrelation of pre-programmed rules, which are iteratively executed with respect to their context-specific objectives.

Every proactive scenario essentially consists of a set of integral rules, which characterise the different aspects of the given proactive scenario, such as its complexity and function orientation. The rules, representing our second-level deterministic principles are characterised as necessary system elements, which allow all MTA and TRG scenarios to function. The secondary deterministic factor has the methodological objective to specify the approach characteristics on a low algorithmic level, which deals with rules' internal specifications and their corresponding structural organisations. In a general perspective, the internal rule's structure is characterised by several computing steps, representing the algorithmic flow of a rule's logic. The rules's integral steps must therefore be executed either sequentially or non-sequentially (intermittently), depending on a type of a confining proactive scenario. Thus, a rule's integral structure consists of a fixed number of successive functions, which are characterised as standard and common for all types of rules. The only slight distinction, which differentiates the rule's structure is defined by rules' functions organisation and their execution. Such distinction fully depends on a type of the proactive scenario, where the given rule is integrated as an algorithmic step of a scenario's logic. Consequently, the rule's compound structure, pertinent to a Meta-scenario differentiates to a certain extent from the rule's structure of a Target-scenario, in terms of its functional composition and the characteristics of the successive execution.

Below, we give an overview, delineating the rule's compound structure and the characteristics of its integral components, which are defined and designed with an objective to highlight the generic character of rule's structure (see Algorithm 1). The generic characteristic of a rule's structure plays an important role in a development process, where it serves to define the rule's universal template, used in the design and development of new proactive rules. The rule's template is therefore characterised as a guiding schema, allowing to create structurally similar rules. From a design-and-development point of view, the rule's template reflects in a general manner, all necessary information-processing and information-parsing prerequisites, which play an important role in abstraction and embodiment of an external context data.

The given template, characterising the rule's compound structure consists of five inter-related parts, or in other words algorithmic elements, which represent the formal steps of rule's information-processing and information-parsing policies. The given algorithmic elements are defined as *Data acquisition*, *Activation guards*, *Conditions*, *Actions*, and new *Rule generation*. All five elements are outlined in their logical and respective positions, where each element, representing a specific function has to sequentially perform its individual tasks and objectives. Consequently, the given approach of fractional data management is characterised by the rule's ultimate objective to operate in relation to its main goal, processing and parsing the corresponding context information.

As we will see later, all rule's elements are characterised as inter-depended and complementary, which implies the rule's correlative structure where every element relies on the state of its above-positioned neighbour. Such rule's characteristic, suggests that activation process of each element requires as a precondition the argument information, which has to be computed first by the rule's preceding neighbour. The given approach allows the context data to be embedded and treated further in a progressive and cumulative manner, which in our opinion represents an optimal and effective solution for monitoring the highly dynamic context conditions.

Algorithm 1: Rule's algorithmic structure

Data: **DA** - Data Acquisition, **AG** - Activation Guards
C - Conditions, **A** - Actions, **R** - Rule regeneration

Data Acquisition()

1: **repeat** for each **DA** request
(a) perform **DA**
(b) **if** (error) **then**
raise exception on system manager's console and go to step 7
else create local variable, initialised to result of **DA**;

2: create local Boolean variable *activated*, initialised to **false**

Activation Guards()

3: **repeat** for each **AG** test
(a) evaluate **AG**
(b) **if** (result == **false**) **then**
go to step 6
else if (**AG** == last **AG** test) **then**
activated=**true**

Conditions()

4: **repeat** for each **C** test
(a) evaluate **C**
(b) **if** (result == **false**) **then**
go to step 6

Actions()

5: **repeat** for each **A** instruction
(a) perform **A**
(b) **if** (error) **then**
raise exception on system manager's console and go to step 7

Rule Generation()

6: **repeat** for each **R**
(a) perform **R**
(b) insert newly generated rule at the end of the queue

7: delete all local variables

8: discard rule from the system

Data acquisition. From a design perspective, the first element of the rule's structure is characterised as a collecting type, which has a functional objective to acquire and to collect all related data, forming the rule's information-specific context. The first rule's element allows the algorithm to gather and to assemble all needed information from a corresponding data source, which can be represented either by the system's sensors or target-system's database. On the other hand, the data assembling process is mainly characterised by the algorithm's objective to collect the context-specific data for its further analysis, which is successively performed by the remaining parts of a rule. As we may notice, the context-specific characteristics of a rule are defined by a type of acquired information, which subsequently specifies the rule's further behaviour.

In order to illustrate the aforementioned details, we provide below an example, which delineates the situation of a new assignment detection on the Moodle platform. Accordingly, if the acquired data by the first rule's element is a new assignment, posted on Moodle, after its detection, the rule's following objectives are to specify its metadata, that is assignment type and its corresponding deadline. The accomplishment of the given objective is achieved by parsing the collected data further through the remaining elements of a rule.

The data acquisition function of the rule gathers all necessary, context-specific information from the Moodle's database. As we mentioned earlier, our objective to choose the given data source is mainly stipulated by the specifications and directives of our implementation approach. More specifically, the aforementioned implementations characterise the Moodle's database as a viable source, capable to reflect the numerous aspects of a user's online activity and to identify the corresponding context changes.

After having successfully retrieved the specified, related information from the Moodle database, the rule has to further save the associated data for its subsequent use by the remaining, successive elements of the rule. The information, acquired from the Moodle database is always saved in a corresponding rule's local variable, which is always available for a later use, if necessary, in the successive information processing. The variable data type can vary depending on the properties of the saved information, where a type of the corresponding information, depends on the rule's main objectives. However, with no difference to its type, a variable is always characterised as read-only, meaning that the given variable can't be modified by a rule nor by any functions.

Consequently, in reference to our example above of new assignment detection, the data acquisition element has to identify first a data change in a corresponding database table, reflecting the posting of a new assignment. Subsequently, the rule's function saves the corresponding data into a boolean type variable, which is parametrised either as true or false. The variable value is later used by the successive rule's elements, which have the objective to specify the various data-related conditions, such as assignment's type and its corresponding deadline. In terms of the information processing steps, the given data acquisition element may be characterised as an opening gate for the successive algorithmic processes. In other words, the data acquisition step has to be executed first, in order to allow the succeeding rule's elements to perform their individual tasks, and thus to fulfil the rule's general objective.

Activation guards. If the first element of *Data acquisition* aims to collect the context-specific information, issued from the Moodle database, the rule's second element of *Activation guards* has the objective to further evaluate, the context conditions of the given data. The evaluation process itself is accomplished based on the computation of a related conditional statement, which in the given example of a new assignment, inherently incorporates the data from the rule's first element. Depending on the rule's objectives, the evaluation data can be of different character and from different sources. The present conditional expression has to evaluate the data either to *true* or *false*, and thus to respectively determine if the element of Activation guards is going to be activated or not. Consequently, the obtained result specifies the further directives and character of the algorithm's control flow. In other words, the given element of *Activation guards* is always performed after the *Data acquisition* part with the one objective, to determine the directives of rule's further algorithmic steps.

Depending on the data evaluation result, there are only two control flow possibilities, which specify if the successive rule's elements of *Conditions* and *Actions* are going to be activated or not. On the one hand, if the element of *Activation guards* is evaluated positively, the subsequent rule's elements of *Conditions* and *Actions* are allowed to be executed, and thus are authorised to continue the progressive flow of the rule's logic. On the other hand, if the element of *Activation guards* is evaluated negatively, the subsequent rule's elements of *Conditions* and *Actions* are skipped in order to move the control flow directly to the last, fifth element of *Rule generation*. Depending on a rule's type and objectives of the inclosing proactive scenario, the control flow of a rule deflects towards the fifth step of the algorithm, where the new instance of the same rule is generated in order to continue the monitoring of the Moodle database. Alternatively, the given rule may generate a different rule, whose objective is to provide an additional service or to collect various, supplementary data, needed for completing the initial scenario's objectives. As we will see later, the given action diversity depends on a logic and objective of the inclosing proactive scenario.

Conditions. The third rule's element, denoted as *Conditions* represents the secondary data evaluation layer, where the algorithm's objective is to investigate in extensive details the context-specific data. As we discussed earlier, the given data is introduced by the first element and initially evaluated by the second algorithmic component. Consequently, in its function, the given third element is identical to the previous algorithmic step of

Activation guards. The main objective of the current element is to further evaluate the context details of the initial data through the computation of corresponding conditional statement.

In relation to our previous example of new assignment detection, the given algorithmic step may for instance evaluate the conditions of the assignment deadline, where it has to identify if the deadline of corresponding student's assignment is less than in 3 days or not. Depending on results of data evaluation, the *Action* part may be activated, and some specific actions may be initiated in order to provide the monitoring of the assignment submission. In the similar manner as in the case of *Activation guards*, the present conditional expression has equally to evaluate the data either to *true* or *false* and to successively store the result into a local, boolean type variable. The only slight difference in comparison to the function of the second rule's element is characterised by the subsequent activation type. In the given step, the result of data evaluation will specify if the algorithmic part of *Actions* is going to take place or not. On the one hand, if the element of *Conditions* is evaluated positively, the *Action* part is going to be activated. On the other hand, if it is evaluated negatively, the *Action* part is skipped in order to move the control flow directly to the last element of *Rule generation*.

Actions. The fourth element of a rule, denoted as *Actions* is an algorithmic part of a rule, which is executed in succession, only if the preceding parts of *Activation guards* and *Conditions* have been evaluated positively. The rule's fourth element has an objective to provide the designated context-specific actions in order to fulfil, entirely or partially, the rule's initial objective. The *Actions* part consists of a set of dedicated instructions that have to be performed in relation to the acquired data, collected by the first rule's element of *Data acquisition*. In certain cases, depending on the rule's characteristics and general objectives of the confining proactive scenario, the *Actions* step can be skipped in order to move the control flow directly to the last, fifth element of *Rule generation*. Such algorithmic overstepping is due to an underlying complexity of involved context-actions, which have to be undertaken in order to effectively respond to an invoking situation.

In the aforementioned cases, the designated actions are performed in succession, through specially dedicated rules, which are generated by the fifth element of an initial rule. However, in more common situations, which invoke a simple set of compulsory actions, the given fourth element of an algorithm is usually always performed. The character of its designated actions may vary depending on the complexity and general objectives of the confining proactive scenario. On the one hand, the designated actions may be characterised as primary and direct, and therefore initiated immediately at the moment of the algorithm's element execution. On the other hand, the aforementioned actions may be designated as the preliminary steps, necessary for the progressive execution of a complex set of instructions in order to complete the global objective.

A function of the given fourth element can be equally illustrated in relation to our example of new assignment detection. Consequently, after the process of new assignment detection, general condition evaluation, and assignment deadline specification, the fourth element of a rule has the consecutive objective to create a message to students, who have not yet submitted their assignments. The content of the message will contain a note, which will notify the students that an assignment deadline is approaching and will be over in three days. The role of the fourth element in the given example is to perform rather a preliminary action of composing the dedicated message to students.

The given approach represents a necessary compound step for completing an entire action, which is an integrated part of the designated proactive service. The primary motivation for an implementation of the given approach is based on a fact that the entire target-action, reflecting a proactive service may be very complex, and thus very difficult to be realised and implemented only by one proactive rule. In the given situations, our methodology of service implementation is mainly based on allocation of the integrated service instructions within several, specially dedicated rules, which in the end form a complex proactive action. Such complex actions are usually assembled in, and represented by a specific proactive service, which often consists of a set of preliminary steps, necessary for accomplishing the initial service-based objectives. The functions of the fourth element are in this context characterised by the rule's type and the general objectives of the confining proactive scenario.

Rule generation. The last step of the algorithm, denoted as *Rule generation*, is a part of the process, which is responsible for the proactive service progression, and thus in 99% of all cases always performed. Depending on a type of a rule, the given fifth element has the objective either to clone itself, or to generate a new rule in order to provide the continuation of a process for accomplishing the initial global objectives on a level of the proactive scenario. In relation to our example of new assignment detection, a role of the given element consists of two main steps of the rule generation.

After having created a message by the preceding fourth algorithmic part, the rule's next step is therefore to direct the control flow to the fifth element, whose objective is to generate a specially designated rule, which will send the created message to the dedicated group of students. When the fifth element generates the new rule, it consequently passes to the given rule the corresponding parameters that include the previously composed message. At the moment, when the messaging rule is generated, it already possesses all necessary information, characterising either the individual students or the target student group. The given data transition is a primary and necessary condition for the rule in order to be able to accomplish its original task.

The second step of the given algorithmic part consists of the successive process of cloning its confining initial rule for creating the continuous effect of a new assignment monitoring. The *self-cloning* approach is an important methodological attribute, which highlights the main principles and importance of a process' continuation paradigm. Successively, the given approach allows the system to ceaselessly monitor the outward situation conditions in order to be always aware about any context modifications.

As we may notice, the rule's mechanism, highlighting our second level deterministic principle is represented as a complex algorithmic structure, consisting of five main steps, which are characterised as the rule's integral functional elements. Every step of an algorithm has therefore its own specific objective, which has to be fulfilled in order to allow the rule's control flow to progress throughout the algorithm's logic. Moreover, every element's objective has to be accomplished separately in order to sustain an independence and autonomy of the rule's different processes. However, the individual functional objectives of each element are interrelated and conditioned by the common objective of a confining rule.

The rule's structure and its logic remains always the same for all types of rules, which additionally serves as the rule's template during the design, development and implementation of various patterns of the proactive system behaviour. The rules differ only in the characteristics of their functional aspects, which are mainly stipulated by the type of a rule and general objectives of its confining proactive scenario.

A rule may be characterised as the progressive algorithmic structure, allowing to create a unified and adapted mechanism for implementing the continuity-based process of a context monitoring. In the upcoming sections, we give several examples, where we aim to describe in greater details the rule's different structural aspects and their functional variations in the frame of two types of proactive scenarios, Meta-scenario, and Target-scenario. Before proceeding to the description of rule's functional characteristics, we first take a look on a mechanism, which is responsible for the actual execution of all implemented proactive rules.

3.2.4 Rules Running System

The highlighted previously rules' structure, represents only one segment of a proactive mechanism, which initially characterises a variation of the system's proactive behaviour.

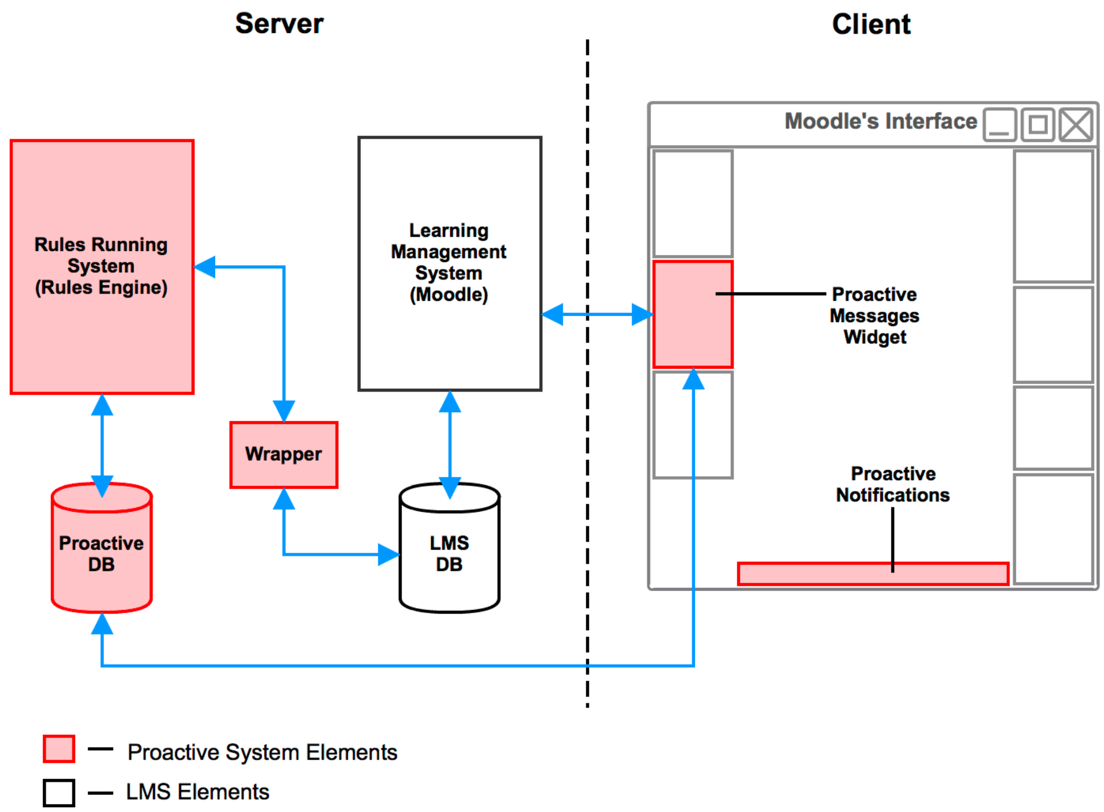

FIGURE 3.1: System architecture

However, individually or by themselves the rules are not capable to perform any proactive actions, as they only represent the logic and a structure of the intended proactive behaviour. In order to implement and to control a rule's logic, which is encoded in rule's functions, the proactive system possesses a mechanism, responsible for the execution of all applied proactive rules [5, 12]. The given mechanism, entitled as *Rules Running System* on a conceptual level represents the core element of the proactive system. In Figure 3.1 we illustrate in the example of the Moodle platform the positioning of the Rules Running System in relation to the different composite modules of the proactive system.

From the general perspective the Rules Running System (RRS) represents a mechanism of the proactive system, which is responsible for the control flow of the rules' execution. Conceptually, the RRS or rules' engine represents the core of the proactive system, which has the unique objective to analyse and to process all composite proactive scenarios. The rule's engine has several essential process parameters, which define the general structure of the RRS functioning. Upon its activation, the rules' engine is set to run for a specific period of time (parameter F), during which it iterates the specified number of implemented rules, stored in the corresponding FIFO list (parameter N). As we may notice, the first two parameters specify respectively the duration of system's activation periods (F), and the maximum number of rules, allowed to be executed within one activation period (N). Additionally, the Rules Running System has the third parameter, which specifies the minimum *pause-time* between two activation periods (parameter P). The last parameter ensures that the rules' engine does not interfere with the main processes of a target-system.

Upon its iteration period, the RRS is programmed to take the first rule from the FIFO queue and to check its compound parameters. Successively, an algorithm of the rules' engine tests the activation guards of a rule and accordingly executes its composite logic. Once executed, the rule is subsequently discarded from the system. For the purposes of sustaining the currently active proactive process, the corresponding rule has to generate a new instance of itself in order to be placed again inside of the FIFO list for its later reactivation.

RRS FIFO lists. Every type of system's proactive behaviour is implemented as a rule in a separate Java class, which is successively stored in the database of the Rules Running System. The RRS is essentially responsible for storing and executing the proactive rules, through the built-in mechanism, which manages the two FIFO lists. The first FIFO list, entitled as *Current queue* is responsible for holding the rules, which are going to be executed in the current activation period. The *current activation period* is consequently defined as an amount of time, during which the RRS is allowed to execute its rules. The given condition is represented as the parameter F.

The second FIFO list, entitle as *Next queue* is responsible for holding the new instances of the proactive rules, which have been generated by the initial rules upon their execution during the current activation period. For the optimisation purposes of managing persistent data, we use the *Hibernate* framework, which allows us to save the queue of to-be-executed rules during each activation period. Furthermore, the given framework allows us to store all corresponding data, relevant to proactive scenarios and the system in general, including context-based generated messages, system statistics and parameters.

RRS First loop. The running period of the rules' engine is initially locked within the main process loop, which can be suspended upon the satisfaction of two conditions. The first condition, allowing the RRS to be stopped depends on a state of the FIFO list, which stipulates that the rule's engine may suspend its operation if the current queue is empty. On the other hand, the RRS can be stopped by an explicit command of the system administrator.

RRS Second loop. An activation period of the RRS is essentially characterised by the engine's iteration process, which conventionally contains the second process loop, responsible for an actual execution of all proactive rules. Similarly to the first loop, the second loop has its corresponding conditions, which accordingly allow it to break the initiated locked process. The given iterating process of the second loop breaks upon several condition: (a) if the current queue gets empty, (b) if the allowed number of executed rules for one activation period reaches its maximum limit (parameter N), or (c) if the allowed time for the current activation period has passed (parameter F). Consequently, upon the satisfaction of one of the three given conditions the RRS iteration period gets suspended.

During the actual period of the engine's iteration, the function processes of the second loop extract the first rule from the current queue and successively execute the logic of this rule by the corresponding implemented method. Subsequently, after completion of the rule's execution, the RRS initiates the next activation period through the several successive steps, including: (1) updating the *current queue* with remaining elements of the *next queue*, and thus emptying the *next queue*, (2) clearing the database cache, (3) replacing the queue on the database with a data of the *current queue*, (4) saving statistics of the completed iteration, and (5) updating statistics of the rule's execution.

Database wrapper. The given composite characteristics of the above modules, collectively represent the overall functional structure of the proactive system. However, in order to connect to a target-system environment, the proactive module has to possess the additional functional elements, which will successively enable a coherent connection between two systems. Consequently, the connection to a target-system is done via its composite database. The proactive system is set to continuously monitor the state of a target-system's database for the purposes of detecting any new changes, which might correspond to a specific logic of one of the proactive scenarios. The given connection between proactive system and a target system is represented by an abstract database wrapper, which includes an implementation of MySQL with the textual SQL queries, needed to access the target system's data.

User interface. The last important element of the overall proactive system architecture is a graphical user interface, allowing to create a direct connection with an outward context environment and its corresponding actors. In the case of the Moodle environment, the interaction with a user is done either by means of the Moodle's email messaging system, or through the implemented user interface elements, allowing to display to a user

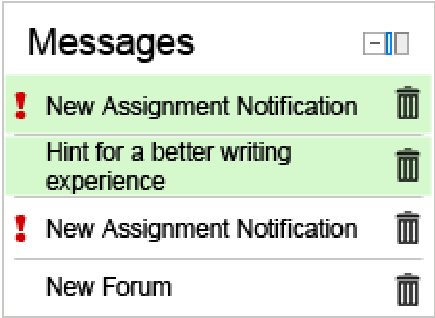

FIGURE 3.2: Messaging block

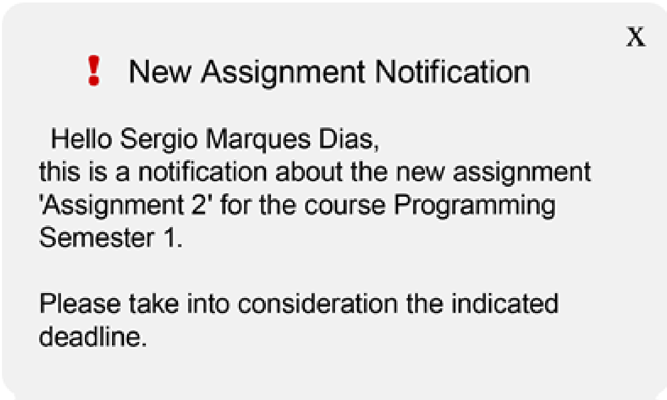

FIGURE 3.3: Message pop-up window

the received notification information via the corresponding messaging windows. Consequently, the user has a possibility to discard the unwanted or read messages by deleting them from the initial list.

The composite elements of the implemented user interface consist of three different parts, including the Messaging block, Message pop-up window, and a Notification bar (see Figures 3.2, 3.3 and 3.4). The Messaging block represents an additional Moodle's module, displayed at the left side of a platform's webpage. The given module contains all messages, arranged into a list, which initially are relevant to, or associated with the user's current e-learning activities. The elements of the Messaging block allow the user a certain degree of interaction with a list, where a message can be clicked, and thus opened, deleted, or simply ignored. A message in the list contains two types of information, that is the title of a message and its corresponding level of importance. Upon the click on a message, the pop-up window is displayed, which allows the user to read all containing message information. In the case when the user is outside of the range of the Messaging block, whenever a new message is sent to a user, the Notification bar pops up in order to get the user's attention. The given last element of the user interface is correspondingly positioned at the bottom of the Moodle's webpage.

As we may notice, the proactive system's functioning is fully dependent on the highlighted above three system modules, the Rules Running System and its database, the database wrapper, and the system's implemented user interface elements. Collectively, all modules represent the proactive system's main framework, which initially ensures a coherent functioning of all applied instances of a system's proactive behaviour. More precisely, the given framework allows us to incorporate the various types of a proactive behaviour, which are correspondingly characterised by the different types of the implemented proactive scenarios. In the upcoming sections we discuss the applied variations and composite features of all implemented proactive scenarios.

3.2.5 Meta-scenarios

Our first type of proactive scenario, denoted as *Meta-scenario* or shortly MTA, represents the system's monitoring capabilities, which are essential preconditions for an effective implementation of a proactive system behaviour. In order to delineate all compound aspects of the MTA-scenario, we explain first the underlying principles and defining conditions related to its name. The name of *Meta-scenario* initially derives from the idea of a meta-concept. In a general perspective, a meta-concept represents a situation where one system process works for the objectives of another process. Consequently, the Meta-scenario's function is defined as a prerequisite process, supporting and completing another related process for the purposes of fulfilling the common goal. A mechanism, based on the given approach is primarily characterised by the sequential and incremental function execution, allowing to create a continuity-effect for the corresponding system processes.

Conventionally, the proactive system consists of a set of predefined proactive scenarios, which constitute the system's modular structure. Every proactive module is a separate system unit, characterised by its own individual objectives and task orientations. Collectively, all proactive modules form a goal-oriented, unified, context-aware mechanism, capable to interact and to monitor the outward context environment. The proactive scenarios, representing different algorithmic processes are interrelated on the low and high levels of the system operation. The given conceptual characteristic presumes that in a global frame of the proactive system operation, one proactive scenario is always working for the objectives of another proactive scenario in order to fulfil the common goal.

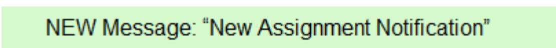

FIGURE 3.4: Notification bar

A Meta-scenario is a prerequisite process, consisting of a set of predefined rules with the specific purpose to provide the associated Target-scenarios with essential context-specific information. In other words, the MTA-scenario is characterised as a modular, sub-level function of the system framework in relation to the higher function of the TRG-scenario. The functions of both types of proactive scenarios are characterised as interdependent, meaning that by themselves, the MTA and TRG-scenarios are only half processes, unable to operate in full capacity, and thus unable to accomplish their original objectives.

By definition the MTA-scenario is a continuous never ending process. In relation to the aforementioned rule's classification, the scenario's compound rule is always subject to the perpetual regeneration with no difference to the values of evaluated conditions. More specifically, if the first two elements of a rule (Data acquisition and Activation guards) are evaluated to a negative value, the rule's control flow will always proceed to the last fifth element in order to generate a new instance of itself. The given rule's characteristics allow the system to sustain for a specified period of time the currently required processes. Therefore, in relation to the MTA-scenarios, the main objective of a rule's regeneration is to provide the proactive system with the context-aware capabilities.

Conceptually, an MTA-scenario has an additional added-value quality, which is defined and characterised as a computationally optimal solution for a cost-effective system operation. More precisely, in comparison to the TRG-scenario, an MTA is the only type of proactive scenario, which is defined as a continuous and never-ending rule. Such methodological distinction implies that all Target-scenarios can be effectively kept backstage of the system process in a non-active state as the latent and waiting system's sub-processes. The given approach characteristics indicate that the system's proactive scenarios don't have to be activated, all at the same time. The few dedicated proactive rules can be constantly in the active state in order to be able to manage and, if necessary, to trigger the single Target-scenarios.

After being activated, the associated rules of a TRG-scenario perform their action-based tasks in a dedicated context environment. Ultimately, the rules of the TRG-scenario get dismissed without regeneration. This is the main distinction between two types of proactive scenarios, where the compound rules of a Target-scenario never regenerate themselves, but are activated correspondingly by a designated Meta-scenario. On the other hand, a compound rule of a Meta-scenario always generates a new instance of itself in order to be able to sustain the process of a context monitoring. Through the conceptual division of all proactive scenarios into MTA and TRG-scenarios, we are able to keep in a non-active, latent state all non-essential and currently needless processes.

Furthermore, we are simultaneously able to sustain all prevailing, compulsory system activities.

The concept of a Meta-scenario allows us to cut a considerable number of the active proactive scenarios, and to consequently provide the cost-effective system operation. The given cost-effective characteristics include the following system optimisation properties: a substantial memory gain, system's information processing discharge, and an effective task delegation mechanism.

In addition to the cost-effective characteristics and system optimisation benefits, an MTA-scenario contributes furthermore to the system process in a different constructive manner, which characterises a Meta-scenario as the necessary context provider. This aspect represents the main objective and functional orientation of an MTA-scenario. Consequently, all implemented Meta-scenarios serve as a perception centre of the proactive system, which continuously monitor the Moodle database for any states' fluctuations.

As mentioned earlier, in order to perform the database monitoring, the main rule of the associated Meta-scenario has to be constantly in the active state, which is achieved through the process of its sequential and continuous regeneration. The given disposition represents furthermore the main precondition for the uninterrupted and never-ending context analysis.

Moreover, the Meta-scenario has a secondary operational objective, which is characterised as a task delegation function. The given function is conceptually based on the rule's first objective of the context monitoring. After having detected the required context data, a Meta-scenario has to provide the associated Target-scenario with the corresponding context information, which will allow the proactive system to successively initiate the necessary context-mediating actions. An MTA-scenario is always able to provide the corresponding Target-scenario with all necessary context information, and thus to accurately respond to an attention-required, context situation.

Moodle-specific Meta-scenarios

In order to exemplify the aforementioned theoretical description of MTA-scenarios in the framework of their practical implementations, below, we present in extensive details the typology and principles, pertinent to all Meta-scenarios. As stated earlier, the main target environment of our empirical implementations is the Moodle platform, which is used by our University[3] as an e-learning, assisting tool. According to the presented scenarios' classifications, we design and develop two types of experimental, proactive

[3] University of Luxembourg

scenarios, which aim in the framework of the present study to reflect the fundamental principles of a proactive system behaviour. The compound rules of both scenario's types are uniformly based on the five-step structure of the rule's template. The orientation of the design characteristics of corresponding proactive scenarios entirely depend on a type of the chosen target-environment. We focus on creating several fundamental categories of Meta-scenarios, which must successively cover the basic dimensions of a context and its corresponding attributes.

Due to the system's conceptual characteristics, which stipulate that the main deterministic mechanism is integrated within the framework of proactive scenarios, we don't have to resort to any additional physical sensors or actuators. According to the stated motives, we choose not to use any physical sensors in our study, but to apply instead the mechanism of the Moodle's database monitoring. The given approach allows us to perform the continuous monitoring and evaluation of the priorly specified context conditions, which are symbolically represented in the database state variations. The mentioned approach, representing the proactive system's context-aware mechanism, is fully capable to satisfy, and to thoroughly reflect our initial study objectives that aim to validate the deterministic structure of the proactive system. Below, we present an outline of the implemented Meta-scenarios, where we describe their general characteristics, compound structures, and their context-affiliated roles.

According to our methodological directives, presented in section 3.1.3, prior to the design of Moodle-based proactive scenarios, we first have to define the corresponding context attributes, which will be involved in the given scenarios. The process of a context analysis necessitates a strong supporting and guiding structure, which will help us to ground the multiple abstract aspects of a context on the stable methodological basis. In order to accurately describe the process of the proactive scenarios' design, we are going to refer to the methodological framework of the earlier established theoretical directives.

Meta-scenario of assignment detection: context attributes A1, A2 The context attribute A1, which emphasises the *interactional* aspects of the Moodle platform is characterised in our study by the users' online interactions, related to the assignment exercises. According to our methodological schema, the given assignment tasks, represent successively the sub-attribute A2, which is defined as a *subject of interactions*, or in other words the conditional reason for the user's interactions.

According to the aforementioned disposition, our two context attributes are structurally interrelated on several concept levels. On the one hand, the initial attribute A1 (user's interactions) highlights a variation of online activities between teachers and students, which are taken into account by the designated proactive scenarios. On the other hand, the sub-type attribute A2 (subject of interactions), consequently specifies the additional details and nature of the corresponding users' interactions. The subject, or topic of users' interactions is therefore represented by a posted assignment, which plays the role of the stipulating condition, specifying the character and type of teacher-student interactions. Our first Meta-scenario will have to address the selected context situations, which involve the tasks of assignment posting, assignment accomplishment and assignment submission. More specifically, the aforementioned Meta-scenario will have to be able to detect and to analyse the newly posted assignment on the Moodle database.

In order to implement the system's proactive behaviour and thus to apply the corresponding context actions, related to the new assignment task, an MTA-scenario has first to detect the newly posted assignment through the continuous monitoring of the associated Moodle's database tables (see Algorithm 2). Our first Meta-scenario, denoted as MTA001, has the objective to detect the new assignment, posted by a teacher, and to subsequently activate a set of related proactive scenarios in order to delegate them their specific tasks. The activated proactive scenarios will continue further the initiated process. Specifically, in the frame of a timeline, the MTA001 scenario has the objective to compare the Moodle's database states and to detect any changes, which successively represent an abstraction of user's actions. All detected data variations, inherently represent a specific event of a real life situation, occurred within the margins of the corresponding target-environment.

After having detected the differences in the database states, the MTA001 will consequently launch the appropriate Target-scenarios in order to allow them to perform their specific context-based actions within the limits of the detected real life situation. The associated action scope of the scenario MTA001, can be characterised as inward related due to its function of the internal database analysis. However, a global objective of the scenario is fully oriented to the user's outward environment. The processes of activation and launching of the associated proactive scenarios together with a delegation of the corresponding context tasks, are entirely stipulated by the types and specifications of the compound scenario's functions.

The MTA001 scenario is built and fully based on the earlier presented rule's template, which represents a general model-structure of the scenario's algorithm. From the programming point of view, the inclosed scenario's algorithm, inherently consists of five rule's elements, where each element has its own specific function.

Step 1: Data acquisition. Upon the first algorithm's step of *Data acquisition*, the rule's function is set to comparatively evaluate the Moodle's database state and to detect any new data, which represents a newly posted assignment on the Moodle platform. In

Algorithm 2: MTA001

```
Input: (N)
1: Data Acquisition()
     boolean newAssignment ← isNewAssignment(N);
2: Activation Guards()
     return newAssignment;
3: Conditions()
     return true;
4: Actions()
   no actions
5: Rule Generation()
     if (activated) then
         assignments[] ← getNewAssignments(N);
         for (each assignment in assignments[]) do
             createRule MTA004(assignmentID, 0);
             createRule MTA002a(assignmentID);
             createRule MTA002b(assignmentID);
             createRule NTF001(assignmentID);
             createRule NTF002(assignmentID);
         nextN ← assignmentID;
         createRule MTA001(nextN);
     else createRule MTA001(N);
```

a process of data evaluation, the rule uses a parametrised function, which by means of a comparative information analysis has an objective to identify every new data instance, appeared within the associated database table. In case when a table is defined as static, containing no new data, the rule is set to regenerate itself through the step five in order to continue the process of the database monitoring. On the other hand, in case when a table is defined as dynamic, all new data is identified, and rule's control flow goes progressively through all successive steps of the algorithm.

The process of a comparative information analysis is based on the provided data-parameter, which represents the last known state of the given database table. The aforementioned data-parameter, denoted as N, is provided by the previous instance of the rule's function, which defines the N-parameter as the last assignment, treated by the given rule in a previous iteration. The role of the N-parameter is therefore to set the ID and time markers on a data in order to always be able to identify the newest items in a database table. Subsequently, after having detected the new data instance in a corresponding database table, the rule saves the corresponding positive value (a value which is set to `true`) in a local boolean type variable, denoted in the given example as `newAssignment`. The aforementioned boolean variable and its value are further used by the successive functions of the current rule.

Step 2: Activation guards. The positive value of the previous *Data acquisition* step allows the algorithm's control flow to successively pass to the rule's next element of *Activation guards* and its corresponding function that represents the first set of conditions' evaluation. Upon the second algorithm's step, the rule's function is set to evaluate the global conditions, related to the acquired data of an assignment task. Upon the given computational step, the previously defined, first-step variable `newAssignment` is evaluated by the rule's current function. According to the previously assigned value of the first-step variable, which is set to `true`, the function of the *Activation guards* sets its own boolean variable to the same value (`true`) as the function of *Data acquisition*. It should be noted that such computational aspects of the rule's second element, which simply repeats the function of the previous step, entirely depend on the type of the variable and data of the corresponding first algorithmic step. Specifically, if the first-step variable is an integer type, the given disposition then naturally extends the possibilities of the successive data evaluations beyond the scope of only boolean values.

The rule's second function is allowed to evaluate the first-step integer value in different directions. If the first-step variable is a boolean type, the rule's second function is thus unable to perform more extensive evaluations, as the only possibility is to re-evaluate the first-step variable to the same value as in the step before. In other words, the positive value of the first-step boolean variable of `newAssignment` is automatically assigned to the corresponding second-step boolean variable of *Activation guards* in order to allow the control flow to pass to the next computational step.

Step 3: Conditions. According to the given scenario type and its basic functional structure, the current algorithmic step does not contain any complex calculations. Conventionally, the given rule's element has the objective to evaluate the extensive, additional conditions, related to a type of the detected data instance of the first-step function. Similarly to the case of the second-step computations, the current function simply sets its own third-step boolean variable to a positive value (`true`) in order to allow the control flow to pass to the next fourth step. The given computational aspect is equally characterised by a first-step variable type and the complexity of the confining proactive scenario, which in the current case does not allow the third-step function to evaluate the data beyond the scope of boolean values. The assigned role of the given algorithmic step in this particular example is limited to the automatic data evaluation with the single purpose, to allow the control flow to pass to the next computational step.

Step 4: Actions. Due to the methodological objectives, which prompt us to use the same rule's template for all categories of proactive scenarios, the programming strategies thus may vary, depending on the objectives of the designed proactive scenarios and their structural complexity. As we demonstrated in the previously described algorithm's steps, certain parts of a rule have sometimes a very little amount of tasks to perform. Moreover, in certain cases, the given fourth rule's step of *Actions*, depending on the complexity and configurations of the confining proactive scenario, is sometimes skipped in order to activate the last fifth element of a rule. The given MTA001 scenario, consisting of specifications, which do not presume the functioning beyond the simple detection of a new assignment, is therefore not programmed to initiate any additional actions, directed to context manipulations. The current fourth step of the MTA001 is going to be skipped by the algorithm in order to proceed directly to the functions of the last fifth element.

Furthermore, it should be noted that according to the conceptual characteristics of all Meta-scenarios, the *Action* part of an algorithm in the present case is used less frequently in comparison to the rule's functions of a Target-scenario. Such programming characteristics are due to the main role of the MTA-scenario, which is mainly characterised by the functions of the context event detection, task delegation, and system process optimisation. As we will see later, the fourth algorithmic step plays an important and vital role in the operation of all Target-scenarios.

Step 5: Rule generation. Upon the last step of the rule's algorithm, the associated functions of the final fifth element are always executed with no difference to the type and complexity of the confining proactive scenario. Consequently, at the algorithm's last step of *Rule generation*, the scenario's fifth-step functions have several assignment-based objectives, where each objective aims to specify the associated characteristics of the event conditions. The first function can be characterised as a precondition step, set to specify the associated details of the detected assignment in order to be able to pass the corresponding data-parameters to the generated proactive scenarios. As we may notice, the function of the details' specification corresponds to our earlier A3 context-attribute (see section 3.1.3), which aims to delineate the characteristics of initial factors of the teacher-student interactions. In other words, if the detected assignment is a subject of interactions (attribute A2), which defines the basis of teacher-student communications, the details about the assignment are therefore our characteristics of a subject (attribute A3) that aim to further specify the corresponding context conditions.

After having specified all necessary assignment characteristics, a function of the algorithm's fifth step will subsequently generate/launch the new proactive rules, accordingly parametrised by the recently specified new assignment details. The additional details are the characteristics of the assignment, which include its name, associated course, and corresponding study program. The newly generated proactive rules may vary in type, depending on the structure and objectives of the confining Meta-scenario. In case of the MTA001, the function of the rule's last step will generate/launch several MTA and TRG scenarios in order to proactively respond to the potential needs and requirements, related to a context situation of the assignment accomplishment.

First, the current rule's function will launch the associated Meta-scenarios, denoted in our study as MTA002a, MTA002b, and MTA004, which will start the continuous monitoring of the given assignment from a different angle. The corresponding MTA-scenarios will monitor the assignment-related context conditions, such as the rate of the assignment accomplishment, performed by students within a particular timeframe. As we may notice, the chosen approach allows us to successively delegate the corresponding context-based tasks to other scenarios in order to continue the proactive monitoring of the specified event from a different contextual angle. The task delegation mechanism is the main technical instrument of the proactive system, allowing to follow, monitor and if necessary to mediate the attention-required context events. In a predominant number of cases, related to the proactive context monitoring, the task delegation is realised and implemented by means of the given fifth step of the new rule generation.

Additionally to the generated Meta-scenarios, a function of the current algorithmic step will subsequently generate several Target-scenarios, parametrised accordingly by the recently specified assignment details. In comparison to the previously generated MTA-scenarios, which aim to take over the process and to continue the context monitoring from a different angle, the generated TRG-scenarios have a simpler objective to provide the single, one-time target-actions. We will discuss the associated functions' details of TRG-scenarios in the following sections.

The provided parameters, assigned to the TRG-scenarios characterise an assignment task by means of establishing its unique identity. A function of the fifth step will therefore generate/launch the associated Target-scenarios and provide them with the necessary parameters, characterising the details of the new assignment. The generated scenarios, denoted in our study as NTF001 and NTF002, will notify the related users about the newly posted assignment. The Target-scenario NTF001 will provide the students with the essential details about the new assignment, whereas the scenario NTF002 will inform the related teacher that all students, enrolled in his/her course are duly notified about their new task. As we will see later, the Target-scenarios occasionally employ an approach of the task delegation. The given disposition is true in certain cases when a TRG-scenario requires an additional service or information, which is necessary for the ample and thorough achievement of its initial objective.

After having generated the associated MTA and TRG scenarios, and thus having successfully delegated all related tasks to the subsequent system processes, the function of the given algorithmic step will ultimately perform its last action. The compound rule of the MTA001 will generate a new instance of itself with the updated parameters of the last treated assignment. The given approach will allow the new instance of the MTA001 to take over the process from its last known state, and thus to continue the monitoring of Moodle's database tables with respect to already happened context events. The provided parameters, assigned to the new instance of the MTA001, reflect the unique characteristics and identity of the detected assignment task. Consequently, when the proactive scenario is regenerated, it will start over the process of algorithmic computations from the first step of *Data acquisition*, and will continue onward towards its last step of *Rule generation*.

As we may notice, the given approach allows the system to iterate the steps of the context monitoring indefinitely, until the process is stopped either by the system administrator or by another proactive scenario. It should be noted that after the negative data evaluation at steps of *Data acquisition* and *Activation guards*, the control flow of the algorithm will directly jump to the last fifth step of the rule in order to generate the new instance of MTA001 scenario. In comparison to TRG-scenarios, the Meta-scenarios always execute the given part of the rule regeneration.

In the illustrative description of the MTA001, we presented the scenario's underlying techniques, related to the compound algorithm's operations and their characteristics. The aforementioned rule's processes represent, in their conventional perspective, the overall mechanism, which defines the methodological strategies of the algorithm's composite steps, valid for all types of Meta-scenarios.

Consequently, the main difference lies within the type of an MTA-scenario, its context-related objectives and its general functional complexity, which collectively define the character, structure, and aspects of scenario's composite functions. However, with no difference to the characteristics, related to the scenario's internal mechanisms, all MTA-scenarios are subjected to follow the aforementioned rule's procedures in order to sustain their functional aspect of process continuity. Every Meta-scenario is explicitly designed to tackle a particular context-based event or a situation, which upon our prior context analysis has been considered and marked either as interesting or as attention requiring.

Therefore, all MTA-scenarios are designed and developed according to the individual characteristics of corresponding context situations and their compound events. According to our design strategies, the given situations and their distinct settings represent individual context elements, which conventionally necessitate the system's continuous proactive mediation. Consequently, the related areas of MTA-scenarios' application may vary considerably, depending on the scenario's objectives, corresponding context characteristics, and the desired context outcomes. In the upcoming sections, we present an exhaustive list of all MTA-scenarios, where we highlight their associated general specifications, objectives and the corresponding technical aspects. Before proceeding to the list of interconnected proactive scenarios, we highlight first in the next section the underlying mechanisms and definitions of the associated Target-scenarios.

3.2.6 Target-scenarios

Denoted as *Target-scenario* or shortly TRG, our second type of proactive scenarios represents the system's action mechanism, which has the primary objective to implement the prior specified target actions within the corresponding context environment. Similarly to the descriptive structure of the MTA-scenarios, in order to delineate all compound aspects of the TRG-scenario, we explain first the underlying principles and defining characteristics, related to its name.

The name of *Target-scenario* is based on an idea of performing the context-specific, target actions with the objective to mediate the conditions of a context environment. In other words, the system aims to target the specific context events on behalf of higher level functions of a corresponding MTA-scenario. The mechanism of proactive scenarios essentially represents an interrelated and mutually dependent process, where all composite units are interconnected on the different system layers. Consequently, we create a network of interdependent processes, where each element is always working in the interests and objectives of another process.

The given disposition is particularly valid in case of the TRG-scenario. According to its conceptual definitions, an TRG-scenario represents an instance of a system process, which always benefits from related prior computations, performed on its account. Therefore, all Target-scenarios are always associated with their corresponding Meta-scenarios, which provide TRG-scenarios with the essential data-parameters, aimed to help them to accomplish their action objectives. In other words, all Target-scenarios conceptually represent the hands of the proactive system, which are subjected to comply with the instructions, issued by a higher level function of a Meta-scenario.

Conventionally, if a Meta-scenario is a prerequisite process, collecting all necessary context data, a Target-scenario is then the intended system's instance for which all corresponding precondition processes have been performed. In other words, if an MTA-scenario is characterised as a modular, sub-level function of the system framework, a TRG-scenario represents a modular, on-level function, which aims to bring the proactive actions into the real life context situations. Besides the given conceptual differences of both types of proactive scenarios, another important distinction, which considerably differentiates their typology, lies within the individual algorithmic characteristics of TRG-scenarios' function longevity.

In comparison to an MTA-scenario, which is a continuous never ending process, an TRG-scenario is characterised as a single, one-time action of the short life span, which ultimately gets discarded after its initial objective is accomplished. In relation to the earlier described rule's convention, the mentioned TRG-scenario's characteristics imply that its corresponding, compound rule is never regenerated at its fifth algorithmic step. Although a Target-scenario may generate other associated proactive scenarios, it can almost never generate a new instance of itself due to the strong adherence to the scenarios' structural policies.

The specified policies allow a scenario to run continuously, through the perpetual rule regeneration, only in case of the Meta-scenario type. The given scenarios' policies are implemented for purposes of the two functional objectives. On the one hand, the purpose is to optimise the system's characteristics in order to reach the ideal cost-effective functioning, whereas on the other hand, the purpose is to separate the scenarios' running processes. The later objective is characterised as an essential element, which constitutes the basis of the proactive scenarios' launching mechanism.

Although the Target-scenario is built on the basis of the same rule's template as any Meta-scenario, its algorithmic structure is slightly different with regard to the execution policy. Despite the fact that a compound rule of an TRG-scenario follows the same five steps of the algorithm's template, it has nonetheless the different functional strategy of the rule regeneration. Moreover, according to the algorithmic structure of an MTA-scenario, we know that the compound rule's step of *Action* is the only part of a Meta-scenario, which is used less frequently, in relation to the rest of its rule's elements. The given algorithmic part however represents a vital functional element in the case of a TRG-scenario.

Consequently, The functional characteristics of a Target-scenario presume less frequent use of its rule's second and third algorithmic steps, which aim to evaluate in a consecutive manner the context-related events' conditions. The given functional characteristics are mainly stipulated by an approach of the process separation, which conventionally divides the common task between two types of proactive scenarios. The associated events' conditions, related to the system's function of a context analysis are always evaluated by a prerequisite process of the Meta-scenario. The rule's second and third steps of an associated Target-scenario are in most cases simply set to the default value (true) in order to allow the control flow to pass towards the next step of *Actions*. Conventionally, the rule's fourth step of a TRG-scenario is almost always performed, as it represents the defining element of a Target-scenario's functional structure.

As previously defined, one of the objectives of the MTA-scenario is to be the *Context provider*, which consequently appoints the given proactive module to serve as a perception centre of the proactive system. Respectively, the main objective of a TRG-scenario can be characterised as the *Action provider*, which impels the given proactive module to initiate the event-specific target actions, oriented to manipulate the context environment. In other words, a Target-scenario uses the context details, provided by an associated Meta-scenario in order to accomplish their common objective as a composite function of a proactive system behaviour.

The aforementioned properties and features of a TRG-scenario, accordingly characterise the given scenario as the hands of the proactive system. The Target-scenario always acts with the objective to provide the targeted context mediation by initiating the direct actions upon the defined instances of a context situation. In comparison to the concept characteristics of a Meta-scenario, the Target-scenario is always subjected to remain in its non-active state, prior to the moment of performing its dedicated tasks. The activation of a TRG-scenario is conventionally assigned to the corresponding MTA-scenarios, which individually decide when to launch the associated Target-scenarios. Additionally, a Meta-scenario provides a TRG-scenario with the corresponding data-parameters, which consequently allow the scenario's algorithm to specify the characteristics of an associated context event. Consequently, if the Meta-scenario represents the primary module of the proactive system, which is capable to capture and observe an event of interest, the Target-scenario represents the secondary module, allowing the system to implement the context-mediating actions.

Moodle-specific Target-scenarios

In order to exemplify the aforementioned theoretical description of TRG-scenarios, below we present in extensive details the typology and principles, pertinent to all Target-scenarios. In the upcoming paragraphs, we present the corresponding detailed structure, which characterises the functional aspects of the second type scenarios.

The presented below structure of a TRG-scenario is built within the guidelines of the aforementioned context attributes. Consequently, all context attributes from A1 to A5 are applied in order to help us to specify all related context conditions and their specific events. The usability of the given methodological framework is pertinent to the design and development of all proactive scenarios. The established methodological framework helps us to define the specific target actions, corresponding to the associated context events, detected by the MTA001 scenario. All Target-scenarios and their specific context actions are designed and developed with the objective to take into account and to employ all associated context data, provided by an MTA-scenario. Every TRG-scenario inherits the characteristics of a context event, which are specified and subsequently provided by the associated MTA-scenario.

At present, we attempt to describe the conceptual and algorithmic characteristics of one of the Target-scenarios used in our empirical study. The given description represents the general underlying principles, common to all TRG-scenarios. In order to implement the system's proactive behaviour, and thus to apply the corresponding target-actions, related to the new assignment task, an MTA-scenario has first to detect the newly posted assignment through continuous monitoring of the Moodle's database.

According to the design characteristics of a proactive behaviour, after having detected the new assignment, and having specified all relevant context details, the system will have to notify the corresponding students about their new task. In order to initiate an event mediation, our MTA001 scenario, from the previous example, will launch the corresponding Target-scenarios in order to respond to the current needs of a context situation. One of the triggered services is a notification scenario, denoted in our study as NTF001. The unique objective of the NTF001 scenario is to promptly notify all related students about their new, course-related task. Notably, the main objective of the NTF001 is divided into several smaller sub-goals, representing respectively the scenario's composite functions. Each function aims to individually fulfil its specific, event-based objective. In order to be able to create the corresponding notification message, the NTF001 has to specify additional context conditions, related to the event, including the assignment name, course name, and the designated group of students. Subsequently, the NTF001 has to generate a new target scenario, denoted in our study as the *Messaging scenario* (MSG001), in order to delegate the given scenario the successive task of sending the already composed and fully specified notification message.

The specifications of a message contain all necessary information about the given assignment and the dedicated group of students. The given processes of the message creation and students' group specification consists of the progressive execution of several compound functions, where each function has to perform its unique action in order to ensure an effective operation of the NTF001 scenario. In comparison to the MTA001, the scope of the NTF001 actions can be characterised as the outward related. The given characteristic is due to the NTF001 functions' orientations and their successive effect on the associated context conditions. Similarly to the MTA001, our NTF001 scenario is built upon, and fully based on the earlier described rule's template, which represents a general, model-based structure, pertinent to all proactive scenarios.

In comparison to the described algorithmic characteristics of the MTA001 scenario, the underlying aspects of the NTF001 control flow delineate slightly different algorithmic properties, which are mainly manifested in the rule's functions and their organisation (see Algorithm 3). The given algorithmic distinction is due to the conceptually different type of the scenario's properties, which conventionally characterise the NTF001 as a single, one-time action. In consequence, certain rule's steps of the NTF001 scenario vary in their functional specifications, compared to the underlying rule's mechanisms of the MTA001 scenario.

Algorithm 3: NTF001

```
Input: (assignmentID)
1: Data Acquisition()
     assignmentName ← getAssignmentName(assignmentID);
     courseID ← getCourseIDfromAssignment(assignmentID);
     courseName ← getCourseName(courseID);
     students[] ← getStudentsFromCourse(courseID);
2: Activation Guards()
     return true;
3: Conditions()
     return true;
4: Actions()
     subject ← createSubject(...);
     text ← createText(...);
5: Rule Generation()
     for (each student in students[]) do
         createRule MSG001(subject, text, studentID);
```

Step 1: Data acquisition. The main conceptual difference, related to the first-step functions of the NTF001 algorithm is stipulated by the typological rule's prerequisites of a Target-scenario in conjunction with the characteristics of its data acquisition process. In other words, in case of our NTF001 scenario, upon the first computational step, the algorithm does not look for the new data, but explores instead the additional, event-based details, related to the already provided information by the MTA001 scenario. The given conceptual divergence, related to the algorithm's functional characteristics, has the key objective to complete the original event-based data with the supplementary context details.

In relation to our example, the compound algorithm of the NTF001 uses the assignment ID, representing the provided data-parameter by the MTA001, in order to further specify the complementary characteristics about the *new assignment* event. The given, additional properties include such event attributes as the *Assignment name* and *Course ID*. Consequently, the obtained *Course ID* has the successive objective to further specify the associated course name and corresponding students' list, containing the identification characteristics of all related users. In a process of the data specification, the rule uses the parametrised functions, which have the composite objective to identify every related data instance, currently present within the associated database tables. The given data specification is performed by means of direct querying of the corresponding Moodle's database tables.

The main conceptual difference between the proactive scenarios lies within the defined characteristics of the rule's control flow. Compared to the MTA001, the first-step functions of the NTF001 are always executed in such a way that the rule's control flow never proceeds directly to the fifth step, but instead follows its logical progression throughout the successive algorithmic steps.

According to the aforementioned rule's characteristics, the earlier defined variations of *static* and *dynamic* database states, are not taken into consideration, and respectively are not applied in the given example of the NTF001-scenario. Instead, the algorithm is programmed to monitor the already existing information within the related Moodle's database tables. All necessary data is always detected upon the rule's first step, and subsequently evaluated by the remaining algorithm's functions of a TRG-scenario. After having collected the necessary data in the corresponding Moodle's database tables, the NTF001 saves all associated values in different local variables, including integer and char data type variables. Additionally, the algorithm creates an array data structure, where it allocates and saves a list of the designated students. All aforementioned local variables and their held values are further used by the successive functions of the current rule.

Steps 2-3: Activation guards and Conditions. Consequently, the *Data acquisition* step allows the algorithm's control flow to pass to the rule's next element of the *Activation guards* and its corresponding function, which represents the first set of the conditions' evaluation. At the level of the algorithm's second step, the rule's function is set to evaluate the global conditions, related to the collected data of assignment specifications. According to the given scenario's type and its action-based functional structure, the algorithmic steps of *Activation guards* and *Conditions* do not contain any complex calculations. Due to the conceptual difference between MTA and TRG-scenarios, the first-step functions of the *Data acquisition* do not require any further condition evaluations as their primary objective is to complete the already evaluated context event of the detected assignment. Conventionally, both rule's elements and their corresponding functions have the objective to set their own second and third-step boolean variables to the positive value (`true`) in order to move the rule's control flow further. The role of the given algorithmic steps, in this particular example of the NTF001, is limited to the automatic data evaluation, which has the only purpose, to allow the control flow to pass to the next fourth computational step.

Step 4: Actions. According to the characteristics of both types of proactive scenarios, we may at present emphasise the most prominent parts of any Meta-scenario. In relation to the MTA-scenario's characteristics, the *Data acquisition* and the *Rule regeneration* represent the rule's main elements of a Meta-scenario, as they allow the system to detect the requisite, context-based data, and to sustain the essential factor of continuity.

On the other hand, in relation to the TRG-scenario's characteristics, the *Data acquisition* and the *Actions* represent the rule's main elements of a Target-scenario, as they allow the system to specify the additional context details about the already detected event, and to define the required context-based actions. Notably, the rule's fourth element of the NTF001 scenario represents an important algorithmic step, which conventionally has the objective to define and to formalise all required further actions. In relation to our example, the compound rule of the NTF001, upon the fourth algorithmic step has to adapt the predefined message template, according to the needs of the context conditions. The rule is successively set to create a subject and the main body of the dedicated notification message by inserting several instances of the previously collected data into the predefined message template. The adapted template will then include the specific assignment name, course name, and the corresponding text message, containing an essential information, highlighting the importance of the given assignment task. The composite, fourth-step functions of the NTF001 scenario are fully based on data, provided by the previous MTA001 scenario and are accordingly parametrised by the first-step functions of the current NTF001 algorithm.

Consequently, by the fourth step of the rule, the NTF001-scenario contains all necessary information, including the assignment name, course name, a full list of the corresponding students, and the ready-to-use body of a notification message. As we may notice, the main part of the NTF001 algorithmic process is characterised as a prerequisite process, allowing to achieve the scenario's objective. In comparison to the MTA001-scenario, the NTF001 aims to assemble and to categorise the already present and known context information into coherent data units in order to push them further to the successive Target-scenario. Notably, the specified and accordingly adapted data of the NTF001 is ultimately used by the successive functions of the newly generated TRG-scenario MSG001, which aims to complete the original process, initiated by the MTA001-scenario.

Step 5: Rule generation. Compared to the previously discussed algorithmic structure of a Meta-scenario, the characteristics of the fifth-step functions of the NTF001 vary considerably, regarding the structural aspects of the fifth-step functions of the MTA001-scenario. Distinctly, in case of the MTA001, the proactive scenario has several objectives, associated with the fifth step of the rule. Upon the execution of the fifth-step functions, the MTA001 is programmed to generate/launch a new MTA-scenario, a new TRG-scenario, and subsequently to self-regenerate in order to sustain the process of context monitoring. However, in case of the NTF001, the proactive scenario has fewer objectives, associated with its fifth-step functions.

The algorithm's fifth element of the NTF001 has the unique functional objective, to generate the dedicated Target-scenario, which will deliver the earlier created and accordingly parametrised notification message to the designated students. The NTF001 algorithm is set to use the earlier created array object, which contains a list of relevant students, defined by the first-step functions of the *Data acquisition* element. The given array, containing a list of participants is created with the objective to specify the identification markers of every related student in order to be able to send him/her the corresponding notification message. Subsequently, the function of the fifth step will generate/launch the supporting Target-scenario, denoted in our study as MSG001, whose objective is to deliver the created notification message to all students.

The list of the message recipients is based on the established, students' identification markers, allocated within an array object. Consequently, the MSG001 contains the following parameters, including the subject of a message, main body of a message, and a student's ID. The prerequisite computations of the NTF001, allow the system to assemble and to categorise all available context information for the purposes of its further processing by the successive MSG001-scenario. In comparison to the MTA-scenarios, which aim to take over the process and to continue the context monitoring from a different angle, the given NTF001-scenario has a simpler objective, to provide a single, one-time assembling-action for the successive context mediation.

In certain degree, the proactive Target-scenarios, similarly to Meta-scenarios employ the approach of a task delegation. The given characteristic is generally true in the specific cases, when a TRG-scenario requires an additional service or information, which is necessary for the effective and thorough achievement of its initial objectives. Our NTF001-scenario has the similar function's characteristics, compared to the compound functions of the MTA001. Notably, the NTF001 has the objective to delegate the corresponding context-based tasks to the associated Target-scenario (MSG001), which is programmed to complete the previously initiated instance of a proactive process.

The given technique is highly effective in cases, when the multiple Target-scenarios have to perform at certain levels the similar tasks. For this reason, or more precisely for the purposes of creating more generic, modular system structure, we decided to segregate certain repetitive processes, which are recurrently used by other system's modules. One repetitive process, used by half of our Target-scenarios is a requisite action of sending the particular message, notification, or the reminder to the dedicated users. The given aspect prompted us to create the designated Target-scenario (MSG001), consisting only of the message providing function, which can be used by any module of the proactive system. The given approach of the progressive task delegation allows us to improve considerably the overall system's operation in terms of its cost-effective values. Additionally, the given approach allows us to effectively optimise our code with respect to the more generic programming methods.

According to the aforementioned algorithm's descriptions, the key difference between MTA and TRG-scenarios, which always remains true, is characterised by the scenario's function of the rule regeneration. With no difference to a type, or main objectives of the confining proactive scenario, an MTA-scenario always generates a new instance of itself at the end of the rule's algorithm. Accordingly, with no difference to a type, or main objectives of the confining proactive scenario, a TRG-scenario never generates a new instance of itself, which consequently characterises a Target-scenario as a *non-regenerative system process*. Such conceptual divergence is true in case of our NTF001-scenario, which at the end of the rule's algorithm simply gets discarded by the system. The given approach characteristics allow us to effectively optimise the running processes of our proactive system in terms of its memory usage and a general cost-effective operation.

Throughout the description of the NTF001, we presented the scenario's underlying techniques, related to a general algorithmic structure and its corresponding functional characteristics. The aforementioned rule's processes represent, in their conventional perspective, the overall mechanism, which defines the operation of the algorithm's compound steps, valid for all types of Target-scenarios. According to the type of a context event, which is primarily detected by a corresponding MTA-scenario, the individual TRG-scenarios may vary in terms of their functional characteristics. Every Target-scenario is designed to provide the most needed, context-based actions in order to directly manipulate, or mediate a particular context event or a situation. Consequently, the related areas of application, pertinent to NTF-scenarios, may vary, depending on the scenario's objectives, corresponding context characteristics, and the desired context outcomes. In the following section, we present, within the framework of the performed empirical study, the overall mechanism of scenarios' connections and their underlying functional interdependences.

3.2.7 Assessment and validation of the deterministic approach

In subsection 3.2.3, we have introduced and subsequently explored the deterministic attributes, which constitute the fundamental grounds of the proactive e-learning system. The theoretical and conceptual descriptions of system's characteristics, allowed us to define the main underlying aspects of the deterministic mechanism of the proactive system, including proactive scenario and rule's structure as key elements. For the purposes of a concise presentation, we have illustrated only one example for each type of the proactive scenario. In the previous sections, we described the composite mechanisms and techniques, which respectively define each type of the proactive scenario. Every presented example conceptually represents a general underlying mechanism, applicable and valid for all related types of proactive scenarios. The compound structure and functional characteristic of one illustrated proactive scenario successively helps us to understand the operational and conceptual principles of the remaining set of proactive scenarios.

In order to see the scenarios' functional interdependencies, applied in the frame of an actual context mediation, we further take a look on the schema of proactive system's processes from the perspective of associated empirical studies. From a standpoint of the real-life implementations, the given overview allows us to identify the potential effects of a proactive system's behaviour on the associated context environment.

Accordingly, we present the general schema of proactive scenarios' interdependencies, which characterise and define the aforementioned principles of the system's modular structure. The schema of the proactive scenarios is successively presented and discussed in the framework of the empirical study, which has been conducted as a part of our global research strategy. From a general point of view, the experiments have the objective to help us to assess the validity and accurateness of the chosen methodologies and theoretical principles, which constitute the basis of our system. At present, we highlight only the structure of experiments, their initial objectives, definitions, and settings. The associated results will be presented and discussed later, in chapter 5.

Before proceeding to the details of scenarios' interdependencies, we define first the corresponding objectives and main organisational points, representing the key aspects of our experiments. The given research project consists of two sets of experiments, where each series of empirical investigations has the objective to test and validate the deterministic and probabilistic parts of the proactive system. The present section highlights the description and characteristics of underlying deterministic principles of the proactive system, the probabilistic principles will be respectively discussed later, in chapter 4.

The fundamental part of the deterministic framework consists of several theoretical and methodological orientations. In order to provide the scientifically rationalised proof of concept for our proactive system, the applied concepts and methods have to be tested and accordingly formalised. Therefore, our first empirical study aims to test and to explore the conceptual structure of the proactive system. The empirical study investigates the issues, related to the software reliability and general system performance.

On the other hand, the conceptual structure of the proactive system, represents collectively the interdisciplinary research efforts, which include the expertise of the associated cognitive science domain. Consequently, the second important objective of the given experiment is to test, explore, and to formalise the applied strategies of the cognitive science domain. The empirical study aims to assess the learning-related aspects, user's perception of various system's functions, and effects of a system's proactive behaviour on a user's overall performance. All aforementioned study objectives are separated into computer and cognitive science perspectives, the *System-related category* and the *Learning-related category.*

System-related category. In our first study perspective of the system-related category, we differentiate two sub-types of the associated study domains, which are subject to the subsequent thorough examination. The first study domain comprises the general character issues, such as functions' variety, their quality, performance characteristics, and ease of software usability. The given enquiry issues are based on a type of the implemented proactive scenarios, their corresponding relationships, and overall running characteristics. The second domain deals with the visual aspects of the proactive system's graphical user interface (GUI), which is embedded into the Moodle's native graphical structure. The proactive system's interface is characterised by the individual GUI elements, position of these elements, effectiveness of their functions, their visual aspects and so on.

Both sub-domains represent collectively the system-related category on a more detailed scale, which allows us to test the individual, constituent elements of the proactive system with regard to general guidelines of a software reliability. In addition to the aforementioned characteristics of the system-related part, we implement furthermore the various types of automatic statistics' collection. The gathered statistical data, aims to highlight the aspects of a user's productivity, user's perception of new functions, and a user's willingness to exercise the implemented proactive system functions. Additionally, we collect various technical statistics, which mostly reflect the characteristics and quality aspects of a system's performance in terms of its cost-effective values.

Learning-related category. In our second study perspective of the learning-related category, we investigate a variation of the system's potential effects on a user's performance in relation to his/her e-learning activity, which may be instigated by the implemented proactive features. Due to the character of the study's initial objectives, a predominant part of the implemented proactive features is appointed to cover the student's platform-based online activity. In consequence, we focus on exploring and studying the aforementioned system's effects on a user's performance, related to the student's e-learning activity. The given part of the experiment aims to study the aspects of users' e-learning behaviour, which includes but is not limited to such phenomena as practice of e-learning activity, cooperative and collaborative learning, development of learning competences, students' general performance, motivation and so on. Conventionally, all aforementioned learning-related phenomena collectively represent an area of interest of our empirical study, which aims to reveal a potential impact of the proactive system behaviour on a student's overall e-learning performance.

It should be noted that the given experiment does not have as the objective to investigate and to study in details every mentioned learning-related phenomena. Instead, the study aims to identify a general disposition, or the likelihood of the potential positive or negative inclinations of a user's e-learning performance. Ultimately, through the analysis and study of the learning-related category, we are able to specify and to define the second aspect of the determinist approach, which at this point, aims to identify the likelihood of either positive, negative, or static effects, caused by the implemented proactive features. The data, issued from the given empirical study, will successively allow us to build more intuitive and elaborate learning-based proactive scenarios, which may extend the capabilities of the proactive system with regard to the accurate reflection and consideration of users' e-learning goals.

Experiments settings. In order to implement and to accomplish all aforementioned experiment's objectives, we first have to elaborate and to adopt a corresponding schema or plan, which will allow us to proceed throughout the organisational and analytical steps of our empirical study. For the given purposes we create a methodology of the experiment, or in other words the structure of all compulsory proceedings and techniques, which consist of specific steps, activities and guiding directives. Consequently, the organisational and structural aspects of the given methodology include the following prerequisite elements: a place of the experiment, time, involved participants, targeted study domains, and the necessary steps for accomplishing the initial objectives.

In relation to the aforementioned procedural aspects of time and place, we decided to set up our experiment at the campus of the faculty of Computer Science and Communications of the University of Luxembourg for a period of one academic semester. The relatively easy access to the University's electronic resources, such as the University's official e-learning platform, is one of the reasons, which prompted us to choose the given academic environment for our empirical studies. Consequently, the chosen environment provides us with valuable technical and administrative possibilities, which allow us to use the Moodle platform as our target system. We are able to implement our proactive module on top of the Moodle platform as an integrative plug-in, allowing to augment the Moodle's basic functionalities with the proactive system's behaviour. The aforementioned implementation characteristics represent an ideal solution for the experiment as the given methodological settings provide us with a valuable opportunity to involve the faculty students as our main study participants. Therefore, the mentioned organisational factors, play an important role in our decision to choose the academic domain of the University of Luxembourg as our main target environment.

After having defined the time and place of experiments, we additionally identify the specific study domains, which will be further subjected to our quantitative and qualitative analysis. In other words, we choose the particular classes and study programmes, which will be treated during one academic semester by the proactive system module, implemented into the Moodle platform. The empirical study covers the activity of two different courses of the bachelor level, given at the faculty of computer science, *Algorithmics 2* and *Probabilities*. The chosen courses include in total 7 assignments, allocated within a period of one academic semester, which students have to accomplish and to electronically submit into a corresponding field of the Moodle platform. The primary objective of our system is to treat the given 7 assignments and their corresponding context situations with the proactive system's functions. The involved system processes consist of continuous monitoring and proactive mediation of the specific assignment events.

According to the character of the chosen environment, the students, enrolled in the associated courses, represent the main target group of our experiments. It should be noted that the each course has the different number of enrolled students. In total, for the course *Algorithmics 2* we have 18 enrolled students, for the course *Probabilities* we have 41 enrolled students. Consequently, for the purposes of our experiment we divide all students in two group, that is *study group* and *control group*. The study group represents the key group, which is the subject to our empirical investigations.

In order to acquire more objective study results, both groups have to be proportionally balanced in terms of the equal number of enrolled high and low performing students, present in each group. For the given purposes, the process of a group formation consists of a categorisation of the students' learning performance, based on their previous grades. We create a list of students, which reflects their previous academic efforts and correspondingly characterises the quality of their achievements. During the second step, we balance both groups with respect to our previously created list, we set up each group with an equal number of high performing and low performing students. Consequently, for the course *Algorithmics 2*, we have 9 students belonging to the control group and other 9 students belonging to the study group. For the course *Probabilities*, we have 20 students, which constitute the control group, and 21 students, representing the study group. The participants of the study group have been assigned to use the Moodle with the integrated proactive features, whereas the control group has been accordingly assigned to use the stock version of the Moodle with its default functions.

The given approach, allows us in the end of experiments to confront and to analyse the data, which on the hand represents the students' reactions and performance, related to, and stipulated by the new integrated proactive features. On the other hand, the second part of the issued data represents the users' performance, related to the use of the stock version of the Moodle. The data, issued from the control group, will subsequently serve as a benchmark for analysing the data of the corresponding study group.

The structure of the current experiments requires an accurate design of the experiment's strategy and the necessary measurement tools that have to be used for the ultimate analysis of the issued results. For the purposes of managing the experiment's progression, we create a strategy plan, where we proportionally allocate all required steps throughout the entire semester. The given plan consists of the composite phases, where each phase contains its specific steps and actions. Ultimately, we have three progressive phases, one leading to another. The first phase includes the preliminary studies of creating two groups of students, one proactive profile for each course, and the first questionnaire that aims to highlight the general user's perception of the Moodle's default functions.

The second phase consists of elaborating and collecting various statistical data, reflecting on the one hand, the characteristics of the proactive module functioning, and on the other hand, the aspects of users' e-learning activities, regarding the use of the implemented new features. During the given phase, we create the second questionnaire that aims to reflect the users' views and opinions, regarding the new proactive functions of the Moodle platform. In the survey we use a multiple-choice, open-ended type of questionnaire, where we provide the students with a possibility either to choose the predefined answers, or to write their own. The final, third phase consists of an ultimate data analysis, including the questionnaires' evaluation, statistical data analysis, and the subsequent comparison of students' academic results, issued from both groups.

The presented characteristics of the first experiment are stipulated by several underlying factors, which conjointly define our empirical study with respect to its main objectives, implementation directives, and requisite organisational steps. The following investigation categories of the system-related and learning-related studies represent two main objectives of the given experiment. The organisational aspects of the present empirical study are characterised by the chosen academic environment of the University of Luxembourg and its associated Moodle platform, which plays the role of the corresponding target system. Our proactive module is set to work in conjunction with the Moodle platform with the objective to provide the related group of students in their daily academic tasks with the assisting functions of the proactive system.

The proactive assistance is mainly characterised by a proactive system behaviour, appointed to mediate the key events, related to various contextual settings of an assignment accomplishment. The main functional objective of the experiment is to provide the proof of concept, aiming to assess the validity and accurateness of the chosen methodologies and theoretical principles, which constitute the basis of the system's deterministic framework. The results, issued from the conducted empirical study aim to help us to advance the conceptual, methodological and technical development of the proactive system towards a new dimension of probabilistic principles. The present empirical study has the objective to provide a stable conceptual and technical framework, which will allow us to move the proactive system's deterministic architecture towards a more intricate probabilistic paradigm. In the frame of the given experiment, we present below an overall structure of a proactive system behaviour, applied within the academic environment of the University of Luxembourg.

3.2.8 Schema of proactive scenarios' interdependencies

For the purposes of our first experiment, we have designed and developed a set of proactive scenarios, divided into Meta and Target-scenarios, which have the objective to implement and to formalise the deterministic part of the proactive system. The deterministic approach of the present study is mainly based on the rule's structure and its progressive algorithmic design. Additionally, the deterministic principles are the basis of a general concept of the selective event's modelling. The given approach implies an individual and segregated modelling of the distinct context events, which are selected, studied and coded with respect to the general interests and objectives of the entire schema of a proactive system behaviour. Most of the modelled context events, represented by the corresponding proactive scenarios, interrelate on a conceptual design level and on a technical application level. The schema of the integrated proactive

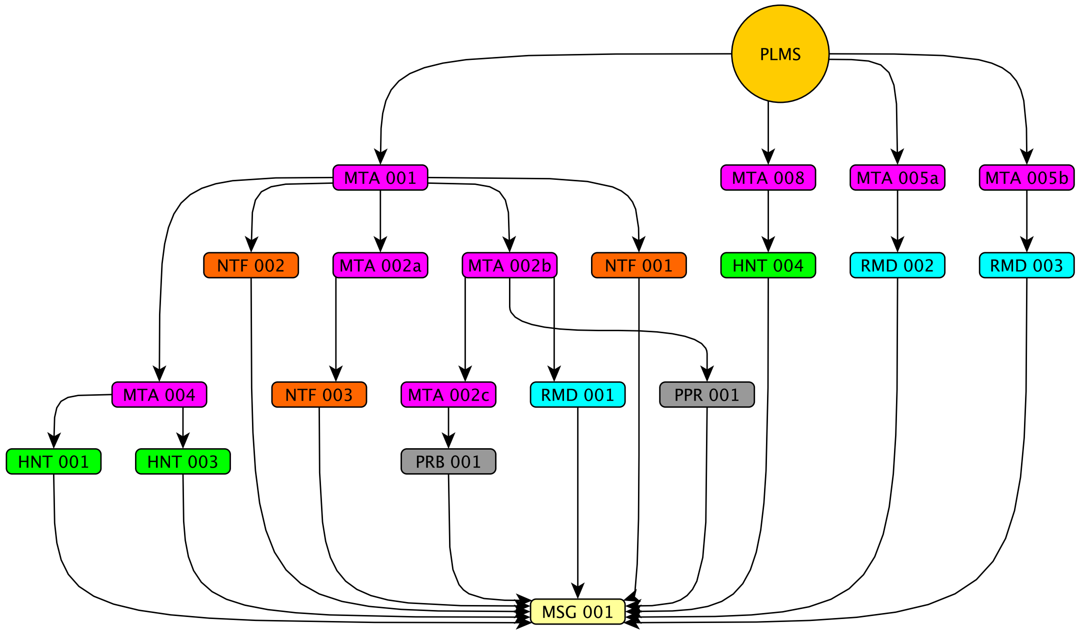

FIGURE 3.5: Proactive process map

scenarios, represents an overall product of all aforementioned deterministic principles, which ultimately constitute the proactive mechanism of a context mediation.

An entire schema of scenarios' interdependencies is based on the continuous operation of 8 MTA-scenarios, which consequently activate, or generate the corresponding sub-processes of TRG-scenarios. The compound rules of MTA-scenarios cover different types of context events. All Meta-scenarios are divided into different categories and activities, including *Assignment detection, Assignment accomplishment monitoring, Assignment-link access detection, Calendar's monitoring*, and *Hint detection* (see Figure 3.5). Correspondingly, we have 12 TRG-scenarios, which are equally divided into different types, according to the characteristics of their activities, including *Notifications, Hints, Reminders*, and *Message provider*. We may notice here the first prerequisites of the process interrelations, such as *Assignment detection–Notification*, *Assignment accomplishment monitoring–Reminder*, or *Notifications–Message provider* and so on. A descriptive list, which highlights all messaging types and their corresponding definitions, used in the given experiment, is presented in Table 3.6.

Given the number and types of the highlighted proactive scenarios, there exists 3 possibilities for the directions of scenarios' interrelations, or in other words characteristics of generated, interdependent processes. The direction of interrelated proactive processes is characterised by the following combinations: MTA–TRG, MTA–MTA, and TRG–TRG. In the first case, the process follows the standard scheme of actions, where an MTA-scenario launches the corresponding TRG-scenarios. It is equally possible that an

Message Type	Message definition
New event notification	System notifies the students about a new assignment.
New event notification for teachers	System notifies a teacher that a notification message has been sent to all students.
Submission notification	System notifies a teacher about the 50% level of assignment submissions.
Writing tasks suggestions	System provides the students with information/hints on how to better perform the writing tasks. The message contains a list of external online sources that are specialised on explanations of planning, structuring, and writing techniques.
Specific assignment hint	System sends to students the hints, provided by a teacher for a particular assignment.
Reminder of an assignment submission	System reminds the students about a deadline of an assignment, if they have not submitted yet their works by the specified time.
Reminder of the upcoming event	System reminds the students and the teachers about an upcoming event, detected on the calendar.
Summary of the upcoming events	System reminds the students and the teachers about the upcoming events of the next week.
Potential problem	System informs a teacher about a potential problem, concerning his/her assignment.
Problem detection	System informs a teacher about an actual problem, concerning his/her assignment.

FIGURE 3.6: Type and definition of a message

MTA-scenario may launch another Meta-scenario, or that a TRG-scenario may launch another Target-scenario. The given characteristic depends entirely on a complexity of the given proactive processes and the main objectives of proactive scenarios. In the case of MTA–MTA direction, an underlying approach is based on the same conceptual characteristics as the rest of proactive scenarios. The general design objectives, allowing to optimise the system operation in terms of a cost-effective functioning, remain valid for all types of proactive processes. The direction of interrelated proactive processes, characterised as MTA–MTA is only applicable in two scenarios, MTA001 and MTA002b. The rest of the Meta-scenarios operates according to their original definition, where an MTA-scenario is always set to generate/launch a corresponding TRG-scenario.

For the purpose of a concise presentation, we name and abbreviate our Target-scenarios in the same manner as MTA-scenarios. The presented below Target-scenarios are labeled and abbreviated in the following manner. All *Notifications* scenarios are abbreviated as NTF, all *Hints* scenarios are abbreviated as HNT, *Reminders* scenarios are abbreviated as RMD, and *Messaging* scenarios are abbreviated as MSG. Additionally, we have two special TRG-scenarios, the first addresses the potential future problems, and is abbreviated as PPR001, the second addresses the already occurred problems, and is abbreviated as PRB001. We proceed further by sequentially presenting all MTA-scenarios, where we give an overview of their compound functions and characteristics of the generated sub-processes.

MTA001. Our first scenario of interest is MTA001, entitled as *New assignment detection*, the given MTA-scenario is set to activate 5 other proactive scenarios, including MTA002a, MTA002b, MTA004, NTF001, and NTF002 (see Algorithm 2). Upon its fifth algorithmic step, the rule activates 3 Meta-scenarios, which are going to continue the process of context monitoring from a different perspective. We discuss the compound functions of the given 3 scenarios later, in the corresponding paragraphs. The main purpose of the present Meta-scenario is to detect a newly posted assignment and to activate a set of subsequent related sub-processes, represented by 3 MTA and 2 TRG-scenarios. Upon the teacher's posting of a new assignment, the system will detect the given event in a corresponding Moodle's database table and will activate a set of additional proactive processes, in order to take care of the associated context events. The scenario follows the standard algorithmic structure of the earlier established rule's template.

NTF001. According to a structure of the MTA001 algorithm's process, the first generated Target-scenario is the NTF001 (see Algorithm 3). The main purpose of the NTF001 is to promptly notify the corresponding students about their newly posted assignment. The NTF001 will use an assignment ID, provided by the MTA001 in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter for the successive MSG-scenario. Notably, the NTF001 will create a notification message, consisting of the predefined body text, assignment name, course name, and the corresponding assignment deadline. The mentioned data is going to be used by a successive sub-process of the message provider. In the end, the NTF001 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to the specified group of related students.

Algorithm 4: NTF002

```
Input: (assignmentID)
1: Data Acquisition()
      assignmentName ← getAssignmentName(assignmentID);
      courseID ← getCourseIDfromAssignment(assignmentID);
      courseName ← getCourseName(courseID);
      creator ← getAssignmentCreator(courseID, assignmentID);
2: Activation Guards()
      return true;
3: Conditions()
      return true;
4: Actions()
      subject ← createSubject(...);
      text ← createText(...);
5: Rule Generation()
      createRule MSG001(subject, text, creatorID);
```

NTF002. According to a structure of the MTA001 algorithm's process, the second generated Target-scenario is the NTF002 (see Algorithm 4). The given TRG-scenario resembles in its characteristics the previous NTF001 scenario. The main purpose of the generated NTF002 is to notify the corresponding teacher about the fact that all students, enrolled in his/her course have been duly informed of a newly posted assignment. The given notification scenario serves as a confirmation for the teacher that his/her assignment has been promptly acknowledged by students. The given scenario will use an assignment ID, provided by the MTA001 in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The NTF002 will create a notification message consisting of the predefined body text, assignment name and the course name, which are going to be used by a successive sub-process of the message provider. In the end, the NTF002 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to the corresponding teacher.

Algorithm 5: MTA002a

Input: (assignmentID)

Data Acquisition()
courseID $\leftarrow$ readCourse(assignmentID);
totalEnrol $\leftarrow$ countEnrolledUsers(courseID);
accomplishedAssignments $\leftarrow$ countSubmitted(assignmentID);
submissionLevel $\leftarrow$ accomplishedAssignments / totalEnrol;
deadline $\leftarrow$ getAssignmentDeadline(assignmentID);

Activation Guards()
return (submissionLevel $\geq$ 50% **and** $<$ 100%);

Conditions()
return true;

Actions()
no actions

Rule Generation()
if (activated) **then**
createRule NTF003(assignmentID);
else if (((deadline == 0 **and** semesterNotOver) **or** (now $<$ deadline)) **and** (submissionLevel $<$ 100%)) **then**
createRule MTA002a(assignmentID);

MTA002a. Our second Meta-scenario is the MTA002a, which is entitled as *Assignment accomplishment of 50%* (see Algorithm 5). The given MTA002a scenario is set to activate one Target-scenario, the NTF003. The MTA002a is consequently activated by our previously discussed MTA001 scenario, with the objective to continue the monitoring process of the already detected assignment. The main purpose of the MTA002a scenario is to continuously monitor the timeline of an assignment submission during one academic semester and, if necessary, to periodically activate a set of subsequent related sub-processes, represented by one Target-scenario. As soon as the system detects that a submission level of the assignment is equal to, or is greater than 50%, it will successively activate the Target-scenario NTF003. The scenario MTA002a follows the standard algorithmic structure of the earlier established rule's template.

Algorithm 6: NTF003

Input: (assignmentID)

1: **Data Acquisition()**

assignmentName ← getAssignmentName(assignmentID);
courseID ← getCourseIDfromAssignment(assignmentID);
creator ← getAssignmentCreator(courseID, assignmentID);

2: **Activation Guards()**

return true;

3: **Conditions()**

return true;

4: **Actions()**

subject ← createSubject(...);
text ← createText(...);

5: **Rule Generation()**

createRule MSG001(subject, text, creatorID);

NTF003. According to a structure of MTA002a algorithm's process, the only launched Target-scenario in the current case is the NTF003 scenario (see Algorithm 6). The main purpose of the generated NTF003 is to notify a related teacher about a level of the corresponding assignment submissions, when it reaches the value of 50%. The present notification scenario serves as the information updater about the status of the related ongoing process. The NTF003 will use an assignment ID, provided by the MTA002a in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The NTF003 will similarly create a notification message, consisting of the predefined body text, assignment name, course name and a value, reflecting the current submission level. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the NTF003 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to the corresponding teacher.

Algorithm 7: MTA002b

Input: (assignmentID)

1: **Data Acquisition()**
  deadline ← getAssignmentDeadline(assignmentID);

2: **Activation Guards()**
  return (now ≥ deadline - 3days);

3: **Conditions()**
  courseID ← readCourse(assignmentID);
  totalEnrol ← countEnrolledUsers(courseID);
  accomplishedAssignments ← countSubmitted(assignmentID);
  submissionLevel ← accomplishedAssignments / totalEnrol;
  return (submissionLevel < 95%);

4: **Actions()**
  createRule PPR001(assignmentID);
  createRule RMD001(assignmentID);

5: **Rule Generation()**
  if (activated) **then**
    createRule MTA002c(assignmentID);
  else if (((deadline == 0 **and** semesterNotOver) **or** (now < deadline)) **and** (submissionLevel < 100%)) **then**
    createRule MTA002b(assignmentID);

MTA002b. The scenario MTA002b, entitled as *Assignment accomplishment of 95%* is a continuation of the process, initiated by the MTA001 scenario (see Algorithm 7). The MTA002b resembles in its characteristics the previous MTA002a scenario. The present MTA-scenario is set to activate 3 other proactive scenarios, RMD001, PPR001, and MTA002c. Upon its fifth algorithmic step, the rule activates one Meta-scenario, which is going to continue the process of an assignment monitoring from a different perspective. The MTA002b is activated by our previously discussed MTA001 scenario, with the objective to continue the monitoring process of the already detected assignment. Similarly to the scenario MTA002a, the purpose of the MTA002b is to continuously monitor the timeline of an assignment submission during one academic semester and, if necessary, to periodically activate a set of subsequent related sub-processes, represented by two TRG-scenarios and one MTA-scenario. As soon as the system detects that by the third day before the due date the percent of accomplished assignments is less than 95%, it will successively activate the Target-scenarios PPR001 and RMD001 in order to mediate the current situation. The scenario MTA002b follows the standard algorithmic structure of the earlier established rule's template.

Algorithm 8: RMD001

```
  Input: (courseID)
1: Data Acquisition()
      assignmentName ← getAssignmentName(assignmentID);
      courseID ← getCourseIDfromAssignment(assignmentID);
      courseName ← getCourseName(courseID);
      dueStudents[] ← getStudentsNotSubmitted(courseID, assignmentID);
2: Activation Guards()
      return true;
3: Conditions()
      return true;
4: Actions()
      subject ← createSubject(...);
      text ← createText(...);
5: Rule Generation()
      for (each student in dueStudents[]) do
          createRule MSG001(subject, text, studentID);
```

RMD001. According to a structure of the MTA002b algorithm's process, one of the generated Target-scenarios is the RMD001. The main purpose of the generated RMD001 is to remind the corresponding students about an approaching deadline, if they haven't submitted yet their assignments by the specified time (see Algorithm 8). The given RMD001 scenario in conjunction with the MTA002b serves as the event reminder about the 3-days-away, approaching deadline of an assignment, which hasn't been submitted yet by certain students. The RMD001 scenario will use an assignment ID, provided by the MTA002b in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The RMD001 will create a notification message, consisting of the predefined body text, assignment name, course name and the corresponding assignment deadline. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the RMD001 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to the specified group of related students.

PPR001. The next Target-scenario, generated by the MTA002b is one of our aforementioned special scenarios, which deals with a potential future problem (see Algorithm 9). The generated proactive scenario, according to a structure of the MTA002b algorithm's process is the PPR001. The purpose of the given Target-scenario is to inform a corresponding teacher about the potential problem, detected by the MTA002b, which

Algorithm 9: PPR001

Input: (assignmentID)

1: **Data Acquisition()**
   assignmentName ← getAssignmentName(assignmentID);
   courseID ← getCourseIDfromAssignment(assignmentID);
   courseName ← getCourseName(courseID);
   creator ← getAssignmentCreator(courseID, assignmentID);
   dueStudents[] ← getStudentsNotSubmitted(courseID, assignmentID);

2: **Activation Guards()**
   return true;

3: **Conditions()**
   return true;

4: **Actions()**
   subject ← createSubject(...);
   text ← createText(...);

5: **Rule Generation()**
   createRule MSG001(subject, text, creatorID);

highlights the unsatisfactory results of assignment's submissions. The PPR001, in conjunction with the MTA002b has the objective to prevent the low submission rate of an assignment, by sending a notification message to a teacher three days before the actual deadline. The given message highlights the fact of a short deadline and provides a teacher with a list of the low performing students. The PPR001 scenario will use an assignment ID, provided by the MTA002b in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The TRG-scenario will create a notification message, consisting of the predefined body text, assignment name, course name, the corresponding assignment deadline and a list of the related students, who haven't submitted yet their assignments. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the PPR001 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to the corresponding teacher.

MTA002c. The scenario MTA002c, entitled as *Assignment accomplishment of 100%* is a continuation of the process, initiated by the MTA001 and MTA002b (see Algorithm 10). The present MTA-scenario is set to activate one additional proactive scenario, the PRB001. The MTA002c is activated by our previously discussed the MTA002b scenario, with the objective to continue the monitoring process of the already detected assignment. Similarly to the scenarios MTA002a and MTA002b, the purpose of the

Algorithm 10: MTA002c

Input: (assignmentID)

1: **Data Acquisition()**
   deadline ← readAssignmentDeadline(assignmentID);

2: **Activation Guards()**
   return (now == deadline);

3: **Conditions()**
   courseID ← readCourse(assignmentID);
   totalEnrol ← countEnrolledUsers(courseID);
   accomplishedAssignments ← countSubmitted(assignmentID);
   submissionLevel ← accomplishedAssignments / totalEnrol;
   return (submissionLevel < 100%);

4: **Actions()**
   createRule PRB001(assignmentID);

5: **Rule Generation()**
   if (!activated) **and** submissionLevel < 100% **and** ((deadline == 0 **and** semesterNotOver) **or** (now < deadline)) **then**
      createRule MTA002c(assignmentID);

MTA002c is to continuously monitor the timeline of an assignment submission during one academic semester and, if necessary, to periodically activate a set of subsequent related sub-processes, represented by one TRG-scenario. As soon as the system detects that one day before the deadline, the percent of the accomplished assignments is not equal to 100%, the algorithm will activate the Target-scenario PRB001 in order to mediate the current situation. The scenario MTA002c follows the standard algorithmic structure of the earlier established rule's template.

PRB001. According to a structure of the MTA002c algorithm's process, the only generated Target-scenario in the present case is the PRB001 scenario (see Algorithm 11). The purpose of the given Target-scenario is to inform a corresponding teacher about the occurred problem, detected by the MTA002c, which highlights the unsatisfactory results of assignment submissions. The PRB001 will use an assignment ID, provided by the MTA002c in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The TRG-scenario will create a notification message, consisting of the predefined body text, assignment name, course name, the corresponding assignment deadline and a list of the related students, who haven't submitted their assignments. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the PRB001 will generate a

Algorithm 11: PRB001

```
Input: (assignmentID)
Data Acquisition()
    assignmentName ← getAssignmentName(assignmentID);
    courseID ← getCourseIDfromAssignment(assignmentID);
    courseName ← getCourseName(courseID);
    creator ← getAssignmentCreator(courseID, assignmentID);
    dueStudents[] ← getStudentsNotSubmitted(courseID, assignmentID);
Activation Guards()
    return true;
Conditions()
    return true;
Actions()
    subject ← createSubject(...);
    text ← createText(...);
Rule Generation()
    createRule MSG001(subject, text, creatorID);
```

new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to the corresponding teacher.

MTA004. Our next Meta-scenario, in a list of the interdependent proactive processes is the MTA004, which is entitled as *Link access detection* (see Algorithm 12). The given MTA-scenario is set to activate two other proactive scenarios, HNT001, and HNT003. The MTA004 is activated by our previously discussed MTA001 scenario, with the objective to continue the monitoring process of the already detected assignment. The purpose of the present Meta-scenario is to detect the first student's access to an assignment link and to activate a set of subsequent related sub-processes, represented by two TRG-scenarios. The system aims to detect the very first student's access to the link of an assignment. As soon as the given event is detected, the algorithm will activate the Target-scenarios HNT001 and HNT003 in order to provide the course-related suggestions to a specified group of the related students. The scenario MTA004 follows the standard algorithmic structure of the earlier established rule's template.

HNT001. The present TRG-scenario is generated according to the algorithm's structure of the MTA004 scenario (see Algorithm 13). The purpose of the HNT001 scenario is to provide a suggestion to the individual students, encouraging them to use the dedicated forums, associated with the ongoing assignments. If the corresponding forums

```
Algorithm 12: MTA004
Input: (N, previousAccess)
Data Acquisition()
    thisAccess ← now;
    courseID ← readCourse(assignmentID);
    assignmentDeadline ← readAssignmentDeadline(N);
    boolean newAccess ← hasFirstAccess(N, previousAccess, thisAccess);
Activation Guards()
    return newAccess;
Conditions()
    return true;
Actions()
no actions
Rule Generation()
    if (activated) then
        createRule HNT001(N, previousAccess, thisAccess);
        createRule HNT003(N, previousAccess, thisAccess);
    if ((deadline == 0 and semesterNotOver) or (now < deadline)) then
        createRule MTA004(N, thisAccess);
```

```
Algorithm 13: HNT001
Input: (assignmentID, previousAccess, lastAccess)
Data Acquisition()
    assignmentName ← getAssignmentName(assignmentID);
    courseID ← getCourseIDfromAssignment(assignmentID);
    courseName ← getCourseName(courseID);
    forums[] ← readNewForums(courseID);
    students[] ← readRelatedStudents(courseID, previousAccess, lastAccess);
Activation Guards()
    return true;
Conditions()
    return true;
Actions()
    subject ← createSubject(...);
    for (each forum in forums[]) do
        currentHeadlines[] ← readHeadlines(forumID);
    text ← createText(currentHeadlines[]);
Rule Generation()
    for (each student in students[]) do
        createRule MSG001(subject, text, studentID);
```

have been found, the algorithm will display the existing headlines of the currently active topics. The HNT001, in conjunction with the MTA004 has the objective to acquaint the students with the currently prevalent topics and the most discussed questions, related to the ongoing assignment. The HNT001 scenario will use an assignment ID, the previous access value and the last access value, provided by the MTA004 in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The Target-scenario will create a notification message, consisting of the predefined body text, assignment name, forum name and a list of currently active headlines. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the HNT001 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to a specified group of related students.

Algorithm 14: HNT003

Input: (assignmentID, previousAccess, lastAccess)

1: **Data Acquisition()**
  assignmentName ← getAssignmentName(assignmentID);
  courseID ← getCourseIDfromAssignment(assignmentID);
  courseName ← getCourseName(courseID);
  students[] ← readRelatedStudents(courseID, previousAccess, lastAccess);

2: **Activation Guards()**
  return true;

3: **Conditions()**
  return true;

4: **Actions()**
  subject ← createSubject(...);
  text ← createText(...);

5: **Rule Generation()**
  for (each student in students[]) **do**
    createRule MSG001(subject, text, studentID);

HNT003. The next Target-scenario, generated by the MTA004 deals with the suggestions, related to the individual writing tasks (see Algorithm 14). The purpose of the HNT003 scenario is to provide the students with an information, or hints on how to better perform in the ongoing writing tasks. The suggestion message contains a list of the external online sources that specialise on the explanation of planning, structuring and writing technics. The HNT003, in conjunction with the MTA004 has the objective to assist the students in their individual writing tasks, which are necessary prerequisites of a successful assignment accomplishment. The HNT003 scenario will use an assignment ID, the previous access value and the last access value, provided by the MTA004 in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The Target-scenario will create a notification message, consisting of the predefined body text, assignment name, course name and a list of the suggested online resources. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the HNT003 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to the specified, individual students.

Algorithm 15: MTA005a

```
Input: (N)
1: Data Acquisition()
     boolean events ← isNewEvent(N);
2: Activation Guards()
     return events;
3: Conditions()
     return true;
4: Actions()
   no actions
5: Rule Generation()
     if (activated) then
         events[] ← getNewEvents(N);
         for (each event in events[]) do
             createRule RMD002(eventID);
         nextN ← getLastEvent(eventID);
         createRule MTA005a(nextN);
     else createRule MTA005a(N);
```

MTA005a. Our next Meta-scenario is the MTA005a, which is entitled as *Calendar's scan* (see Algorithm 15). The given MTA-scenario is set to activate one proactive scenario, RMD002. The MTA005a, equally to the MTA001 is activated at the start of the proactive system. The purpose of the present Meta-scenario is to continuously monitor the Moodle's calendar in order to detect all upcoming events and to activate a set of subsequent related sub-processes, represented by one TRG-scenario. The system aims to detect the upcoming events, basing on an information, extracted from the Moodle's specific database table. As soon as the given event is detected, the algorithm will activate the Target-scenario RMD002, which will further specify all necessary information, required for creating and sending a notification message to the related users. The

MTA005a uses as the initial parameter the last event, which has been treated by the proactive system in the previous iteration. An upcoming event is defined in the given case as an event, which takes place one day before the due day. The scenario MTA005a follows the standard algorithmic structure of the earlier established rule's template.

Algorithm 16: RMD002

```
Input: (eventID)
1: Data Acquisition()
     eventName ← getEventName(eventID);
     courseID ← getCourseIDfromEvent(eventID);
     courseName ← getCourseName(courseID);
     students[] ← getStudentsFromCourse(courseID);
     teachers[] ← getTeachersFromCourse(courseID);
2: Activation Guards()
     return true;
3: Conditions()
     return true;
4: Actions()
     subject ← createSubject(...);
     text ← createText(...);
5: Rule Generation()
     for (each student in students[]) do
         createRule MSG001(subject, text, studentID);
     for (each teacher in teachers[]) do
         createRule MSG001(subject, text, teacherID);
```

RMD002. The current TRG-scenario is generated according to the algorithm's structure of the MTA005a (see Algorithm 16). The purpose of the RMD002 scenario is to provide a reminder notification to the corresponding students and teachers about an upcoming event, which has been detected on the Moodle's calendar. The RMD002, in conjunction with MTA005a serves as a reminder to students and teachers about a one-day-away, course related event. The RMD002 scenario will use an event ID, provided by the MTA005a in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The Target-scenario will create a notification message, consisting of the predefined body text, event name, course name and the corresponding event data. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the RMD002 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to a specified group of related users.

Algorithm 17: MTA005b

Input: (N)

1: **Data Acquisition()**

2: **Activation Guards()**

boolean weeklyEvents ← false;

if (now == time(Friday, 4pm) **and** timeStart(eventID) ≥ nextMonday()) **then**

weeklyEvents ← isUpcomingEvents(N);

return weeklyEvents;

3: **Conditions()**

return true;

4: **Actions()**

no actions

5: **Rule Generation()**

if (activated) **then**

events[] ← getUpcomingEvents(N);

createRule RMD003(N);

nextN ← eventID;

createRule MTA005b(nextN);

else createRule MTA005b(N);

MTA005b. The scenario MTA005b, entitled as *Calendar's scan* is a similar process in comparison to the previous MTA005a scenario (see Algorithm 17). The present MTA-scenario is set to activate one additional proactive scenario, RMD003. The MTA005b, equally to the MTA005a and MTA001 is activated at the start of the proactive system. Similarly to the scenario MTA005a, the purpose of the MTA005b is to continuously monitor the Moodle's calendar in order to detect all upcoming events. In comparison to the MTA005a, the difference lies in the objective of the present scenario. The MTA005b aims to create every Friday at 16:00 a summary of all upcoming events, scheduled for the next week. The MTA005b notifies the related users about the important events before the weekend. The given reminder has the objective to eventually motivate the related students to use their weekend time more productively. Consequently, every Friday at 16:00, the algorithm will create a list of the next-week events and will activate the Target-scenario RMD003, which will further specify all necessary information, required for creating and sending a notification message to the related users. The MTA005b uses as the initial parameter the last event, which has been treated by the proactive system in the previous iteration.

RMD003. The current TRG-scenario is generated according to the algorithm's structure of the MTA005b scenario (see Algorithm 18). The purpose of the RMD003 scenario

Algorithm 18: RMD003

```
Input: (events[])
1: Data Acquisition()
     events[] ← getUpcomingEvents(N);
     for (each event in events[]) do
         eventName ← getEventName(eventID);
         courseID ← getCourseIDfromEvents(eventID);
         courseName ← getCourseName(courseID);
         students[] ← getStudentsFromCourse(courseID);
         teachers[] ← getTeachersFromCourse(courseID);
2: Activation Guards()
     return true;
3: Conditions()
     return true;
4: Actions()
     subject ← createSubject(...);
     for (each event in events[]) do
         currentEvent[] ← readEvent(eventID);
     text ← createText(currentEvent[]);
5: Rule Generation()
     for (each student in students[]) do
         createRule MSG001(subject, text, studentID);
     for (each teacher in teachers[]) do
         createRule MSG001(subject, text, teacherID);
```

is to provide a reminder notification to the corresponding users about the upcoming events of the next week, which have detected on the Moodle's calendar. The reminder notification is sent to the users every Friday at 16:00. The RMD003 scenario will use an event ID, provided by the MTA005b in order to create and to specify all necessary pre-requisite data, which is going to be used as a parameter by the successive MSG-scenario. The Target-scenario will create a notification message, consisting of the predefined body text and a list of the next-week events. Conjointly, all data is going to be used by a successive sub-process of the message provider. In the end, the RMD003 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to a specified group of related users.

MTA008. Our last Meta-scenario is the MTA008, which is entitled as *Specific hint detection* (see Algorithm 19). The current MTA-scenario is set to activate one additional proactive scenario, HNT004. The MTA008 scenario is activated at the start of the

Algorithm 19: MTA008

```
Input: ()
1: Data Acquisition()
     Date now ← getCurrentDateTime();
2: Activation Guards()
     return hasHintsToSend(now);
3: Conditions()
     return true;
4: Actions()
   no actions
5: Rule Generation()
     if (activated) then
         hints[] ← getHintsToSend(now);
         for (each hint in hints[]) do
             createRule NTF004(hintID);
     createRule MTA008();
```

proactive system. The purpose of the present MTA-scenario is to detect the specific hint, provided by a teacher in relation to the ongoing assignment task. The MTA008 monitors the corresponding Moodle's database table for any data, corresponding to a new assignment hint, posted by a teacher. As soon as the new data is detected, the algorithm will activate the Target-scenario HNT004, which will further specify all necessary information, required for creating and sending a notification message to the related users. The scenario MTA008 follows the standard algorithmic structure of the earlier established rule's template.

HNT004. The current TRG-scenario is generated according to the algorithm's structure of the MTA008 scenario (see Algorithm 20). The purpose of the HNT004 scenario is to provide the related students with the previously created assignment hint. The suggestions are provided by a teacher for the currently active assignment tasks. The HNT004, in conjunction with MTA008 has the objective to assist the students in their individual assignment-related activities. The HNT004 scenario will use a hint ID, provided by the MTA008 in order to create and to specify all necessary prerequisite data, which is going to be used as a parameter by the successive MSG-scenario. The Target-scenario will create a notification message, consisting of the specified body text, provided by the teacher. Consequently, the data is going to be used by a successive sub-process of the

Algorithm 20: HNT004

Input: (hintID)

1: **Data Acquisition()**

    assignmentID ← getAssignmentFromHint(hintID);
    subject ← getTitleFromHint(hintID);
    text ← getTextFromHint(hintID);
    courseID ← getCourseIDfromAssignment(assignmentID);
    students[] ← getStudentsNotSubmitted(courseID, assignmentID);

2: **Activation Guards()**

    return true;

3: **Conditions()**

    return true;

4: **Actions()**

    setSentToHint(hintID);

5: **Rule Generation()**

    for (each student in students[]) **do**
        createRule MSG001(subject, text, studentID);

message provider. In the end, the HNT004 will generate a new instance of the accordingly parametrised MSG001 scenario, which will deliver the fully adapted notification message to a specified group of related students.

MSG001. Our last Target-scenario is MSG001, which is entitled as *Message provider* (see Algorithm 21). The purpose of the given TRG-scenario is to provide the system with required messaging functions. The MSG001 is generated and used by all aforementioned Target-scenarios. The MSG001 always uses the parameters, provided by a corresponding Target-scenario, including a message subject, text, and user ID. As a prerequisite action, the MSG001 verifies first the user's status (online/offline). Depending on the obtained result, the scenario either displays a pop-up message box, containing the corresponding notice, or simply sends a notification email. The current MSG001 represents the most frequently used Target-scenario, which is correspondingly characterised as an effective instrument of the direct context mediation. In comparison to all other TRG-scenarios, the MSG001 scenario always plays the culminating role in the earlier initiated proactive processes.

At this point we may notice the interrelated, interdependent and semi-autonomous character of the proactive system functioning, which has been earlier mentioned in our theory chapters. The given approach of the scenarios' structural interdependencies, reflects at

Algorithm 21: MSG001

Input: (subject, text, UserID)

1: **Data Acquisition()**
  boolean isOnline ← checkUserStatus(UserID);

2: **Activation Guards()**
  return true;

3: **Conditions()**
  return true;

4: **Actions()**
  displayMsgBox(subject, text, UserID);
  if (currentStatus = false) **then**
    sendEmail(subject, text, UserID);

5: **Rule Generation()**
no actions

certain degree the aforementioned Tennenhouse's premise of a system's high frequency operation.

Furthermore, we may notice that all presented proactive scenarios, respectively align with our earlier established schema of context attributes (see section 3.1.3). Notably, throughout the examples of scenarios' interrelations we saw that all context attributes from A1 to A5 are represented on different context levels by various instances of a proactive system behaviour. If an assignment submission, represented by the MTA001 is the cause for users' interactions on Moodle (attributes A1, A2), the rest of proactive scenarios respectively represent our context attributes from A3 to A5.

The highlighted proactive scenarios represent the mentioned attributes on different context levels. On the one hand, they aim to characterise an assignment through such information as the assignment name, assignment deadline, assignment-related course, assignment-related students, assignment-related tasks and so on. On the other hand, the aforementioned proactive scenarios have the objective to address and to define all assignment-based activities, including the individual writing tasks, group-based accomplishment of required exercises, monitoring of a user's general performing activity and so on. The given design aspects match our schema of the defined context attributes, including the attributes A3, characteristics of a subject, A4, subject-based activities, and A5, characteristics of subject-based activities. The presented above proactive scenarios have been designed and developed according to the earlier established framework of context attributes. The given approach allowed us to specify and to further elaborate the most important and relevant context events for our empirical studies.

3.2.9 Summary

In the current section we addressed our earlier introduced Research Questions #2. The given research direction aims to investigate the underlying principles and key elements, which constitute the proactive system's framework and its integrity on a fundamental deterministic level. The system's deterministic approach consists of several essential elements that conjointly represent its methodological structure. Notably, the given elements are characterised, on the one hand, by a proactive scenario and, on the other hand, by the scenarios' compound rule and its algorithmic structure. According to our initial research objectives, the developed deterministic mechanisms of the proactive system have been implemented and tested within the academic environment of the University of Luxembourg. We use the University's electronic platform Moodle in order to implement and to ultimately validate the underlying principles of the system's deterministic approach.

By employing the Moodle platform as our target system, we had the initial objective to amplify its basic functions by the integrated features of the implemented proactive module. Among several reasons, the Moodle's framework capabilities have played an important role in choosing the given LMS environment as our target system. The Moodle's framework structure allowed us to build and to implement our proactive module as a host system's plug-in. Consequently, the given methodological aspect allowed us to enhance the Moodle's fundamental functions with a proactive type of behaviour, implemented through the specially designed proactive scenarios.

The modelling approach represents the first prerequisite of our deterministic framework, where we individually design every variation and type of a proactive system behaviour. The created behavioural instances may cover only the specifically chosen areas, settings, and events of a predefined context. In order to implement the system's proactive aspects, we need the appropriate methodological mechanisms, which have to reflect the initial characteristics of the proactive computing principles. For this reason, we design a mechanism of the proactive interdependent processes, which is mainly represented by various proactive scenarios and their compound rules. The scenarios' composite rule is defined as a uniform structure of progressive and interdependent algorithmic steps.

Consequently, our second prerequisite of the deterministic framework is represented by a rule's structure that plays the role of an algorithm's template. All designed proactive scenarios are conventionally based on the given rule's template, consisting of five algorithmic steps. Every step has its own predefined objective, which further defines the character of scenario's build-in functions. Depending on a type and objective of a proactive scenario, the functions of a rule may vary considerably. The key difference between associated rule's functions lies within the rule's fifth algorithmic step. In case of a Meta-scenario, the compound rule always generates a new instance of itself in order to sustain the factor of continuity. On the other hand, in case of a Target-scenario, the compound rule only launches the additional sub-processes and ultimately gets dismissed without any further regeneration. The given rule's characteristics represent the main methodological difference, which accordingly distinguishes two types of proactive scenarios.

The designed and implemented proactive scenarios, representing the fundamental functional mechanism of the proactive system, are subject to a successive testing and formalisation. Consequently, the main objective of our first experiment is to assess and validate the underlying methodological and theoretical principles of the proactive system's deterministic framework. The realisation of the experiment consists of the successive implementation and testing of all proactive scenarios within the framework of the University's electronic platform Moodle. The chosen context characteristic for our first experiment is the learning-related setting of an assignment accomplishment. The domain-specific proactive scenarios have the objective to cover the most relevant context situations, unfolded from the root event of an assignment detection. According to the experiment's initial objectives, all presented proactive scenarios are based on, and designed with the governing objective to assess and to evaluate the validity and reliability of the proactive system's deterministic principles.

The main objective of the present section is characterised by a methodical description of our first empirical study, which aims to ground the conceptual and theoretical principles of the currently employed deterministic framework. The given study objective represents in fact a consecutive, strategic step of our progressive research methodology. The elaborated principles of the deterministic framework are subsequently used in our study as a conceptual basis for our next research phase of designing and implementing the stochastic principles of the succeeding probabilistic framework. Therefore, in order to proceed towards more complex algorithmic structures, we first had to assess and to validate the fundamental deterministic principles of the proactive system framework. The given approach allows us to use the previously tested deterministic concepts for building the more intricate mechanisms of a stochastic data detection and evaluation. The formalised principles of the deterministic approach represent the basis of our successive probabilistic framework. In the following chapter, we present the methodology of our next research step, which aims to delineate the underlying principles of the system's probabilistic mechanisms.

Chapter 4

Probabilistic methodological framework

In the present chapter we define all notions and concepts used in the design and development of the proactive system's prototype. The main objective of the following sections is to situate the methodological implementations within the previously defined theoretical framework. This will help us to scientifically rationalise all related aspects and characteristics of the system's framework with respect to mainstream theories and approaches.

In the following sections we proceed in accordance with our aforementioned research questions (see chapter 1). If the previous chapter has been characterised and built according to the properties of the Research Question #2, then the Research Questions #3 will respectively define the structure of the present chapter.

Correspondingly, in section 4.1, we present the probabilistic dimension of the proactive system, which has been defined and implemented in the framework of Web Search Engines. The main objective of the corresponding section is to delineate and to validate the principles of a probabilistic approach through our exploratory study. Consequently, we highlight the methodology of our second experiment, which aims to enhance the fundamental deterministic structure of the system by more elaborate and incisive concept-features of the probabilistic approach. We present in section 4.2 the details of our exploratory study, which highlight the probabilistic aspects of a proactive inference and evaluation of users' cognitive states during an online search activity.

4.1 Probabilistic approach: proactive computing in the framework of web search engines

The main objective of the present section is to address and to successively investigate the underlying aspects of our Research Question #3, which aims to highlight on an empirical level the probabilistic side of the proactive system. In the current section we present the details of the system's probabilistic methodology, which has been practically implemented and tested within the framework of Web Search Engines. Our objective here is to delineate the fundamental concept principles of a stochastic approach and to validate it through subsequent experiments. We present below the second part of the system's proof of concept. The given section aims to explore the probabilistic side of the proactive system through an empirical study of enhancing the basic search engine functionality with user-oriented, proactive, context-aware capabilities.

4.1.1 Search engine environment

Before proceeding to the description of methodological principles, related to the probabilistic approach, we have to characterise first the underlying nature of our target-environment. The main operational environment of a proactive system is defined entirely by a type of the target-system. Hence, the basic methodological principles, characterising the probabilistic side of the proactive system are going to be applied and tested in the framework of Web Search Engines (SE).

For the purposes of a more adapted and effective application of our approach, we have to thoroughly understand all historical and evolutionary traits, which form the underlying principles of the current search engines' paradigm. The historical overview allows us to identify and to characterise the target-system's strong and weak points of its functional principles. Consequently, during a process of the search engine's enhancement, the identified system's characteristics will allow us to take the advantage of its strong points and to respectively reinforce its weak points. Such a discriminative, target-system analysis will ultimately allow us to design the adapted, domain-specific proactive system behaviour.

For the purposes of our exploratory study, we have chosen the search engine environment as the primary target domain for deploying and testing the probabilistic framework of our proactive system. The chosen type of the target environment is conditioned by several objective-supporting reasons. Notably, the given target-system type, in our opinion, provides the perfect empirical conditions, where a probabilistic estimation of data can be effectively implemented on several design levels. We choose the SE environment due to the characteristics of a target software system, which inherently comprise an extensive user's activity, representing a valuable context data for our proactive module.

A web search engine is a dedicated software system, allowing to perform the targeted search within the vast domain of internet resources. In a conventional interpretation, the targeted search may be characterised as a goal-specific querying of various types of an online information, including web pages, media files, text files and so on. Moreover, depending on the type of the employed search engine and a variety of the searchable information, the data querying may be equally performed within an associated local database. Therefore, we may ultimately define a search engine as an online, or a standalone information mining tool, allowing the user to query the corresponding distributed data.

The history of search engines goes back to the early 1990s, when the first two information indexing tools, appeared respectively in 1990 and 1991. The earliest search engines, allowing to query the already existing handful of internet resources were the software tools, called *Archie* and *Gopher*. The *Archie* search engine was the very first indexing tool, created in 1990 by Alan Emtage, Bill Heelan and J. Peter Deutsch, who at that time were the computer science students at the McGill University in Montreal [76]. The program operated as an indexer of collected files names, which have been retrieved by fetching the directory listings of anonymous FTP (File Transfer Protocol) files in the Internet [77]. A year later, in 1991, Mark McCahill from the University of Minnesota, has developed another information indexing tool, called *Gopher*. The main difference between two programs lied within *Gopher's* functional orientation to index plain text documents. Due to the *Gopher's* text orientation, most of its *sites* became *Web sites* after the creation of the World Wide Web [76]. Thus, the very first prototypes of modern search engines mainly consisted of 3 function components, that is index browser, indexer and search engine itself. The basic function of a single string querying in *Archie*, later on encouraged the development of successive search engines, such as *Veronica* and *Jughead* [78].

The rapid development of information technologies has ultimately led to an evolution of search engines towards more complex, user-oriented paradigms. The new technologies began to progressively address more intricate aspects of human-computer interactions, which started to include such aspects as the adapted search results' recommendations, evaluation of a user's intent, modelling of the user's search patterns, and so on. The scholars from Yahoo! research lab, basing on the analysis of a user's browsing behaviour, propose in their work the mechanisms, allowing to estimate the document relevance by building and employing the assumption models of a user's online activity [79]. The given mechanisms are often based on user's activity models, created through the preceding techniques of collecting and analysing the user's search-related data [80].

In another related work, in order to increase the accuracy level of an information retrieval, Wang et al. proposes a user motivation model, which is based on the analysis of user's behavioural patterns [81]. The authors' assumption is stipulated by the fact that user's search objectives and motivations may lead to the corresponding behavioural patterns, which can be tracked, analysed and successively implemented. Another notable example demonstrates a model-based methodology of creating a computational cognitive model, which effectively simulates a user's search activity, including the navigation and results' selection. The given model has been designed in the framework of the already mentioned ACT-R cognitive architecture, which aims to simulate and ultimately approximate the behavioural characteristics of real users [82]. All aforesaid examples demonstrate an increasing rate of research efforts, directed towards designing new methodologies, which on their conceptual level try to fuse the user-specific cognitive and behavioural aspects into an algorithmic level of the system's computational processes.

However, on their fundamental operational level, the predominant part of currently available search engines (SE) may be characterised as elementary, information querying software tools, which do not take into consideration the aspects of user's cognitive variations during the search [83]. The given software tools are often based on a decontextualised examination of their index databases, which ultimately serves to provide a listing of the best information matching, representing the search results. Furthermore, the process of an information matching and information retrieval is build upon the schema of results ratings, which represents the key mechanism, employed in the identification of relevant data [84, 85].

The fundamental search engine's algorithms usually do not take into consideration a variation of the user's manifested cognitive states. However, the manifested instances of the user's inner cognitive activities and his/her overt behavioural patterns are considered in our study as crucial elements in the identification of the data relevance. In our opinion, in order to increase the relevance of search results, the aforementioned aspects have always to be linked to the process of information querying, and thus to be purposely expressed in a structure of the search engine's matching algorithm.

The chosen type of the target search engine for the present experiment, resembles in its functional characteristics the fundamental elementary types of the aforementioned search engines. Thus, we use a selection-based, standalone version of an online search engine, which operates upon the locally stored database. The database itself is an XML file, containing various entries of the medical terminology, accessible through the search

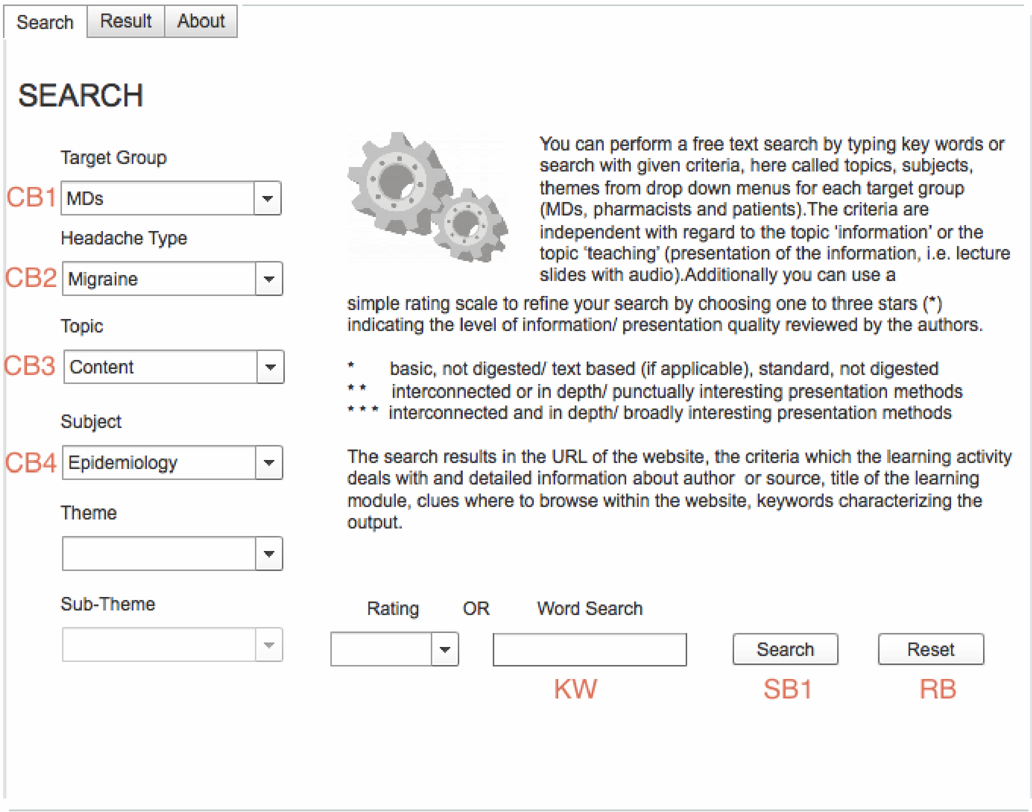

FIGURE 4.1: Search engine, search page perspective

engine's graphical user interface (GUI). The data from an XML file is subsequently translated into the MySQL database management system, which is used by the proactive module for the purposes of data detection.

The interface of our search engine originally contains two visual perspectives, including a page of the search criteria selection, and a page of retrieved results. The search page is a category-based, hierarchically structured representation of the predefined and accordingly arranged topic information (see Figure 4.1). It contains the query data, allocated into logical, topic-based categories, represented by, and accessible through multiple combo boxes. The user may invoke a search query of the medical terminology through the mouse activity (clicks) by sequentially selecting the searchable topic categories in the corresponding combo boxes. The combo boxes operate in a successive order by displaying the categorised topic suggestions, which are essentially based on the context of user's previous selections. Additionally, the search page contains an optional keyword field, which may be used in conjunction with the selected topic categories. The user may optionally type a sequence of characters in order to elaborate an initial category selection.

The result page is a simple interface structure, containing two main visual elements, a list box and a message box (see Figure 4.2). The list box contains the headings of retrieved

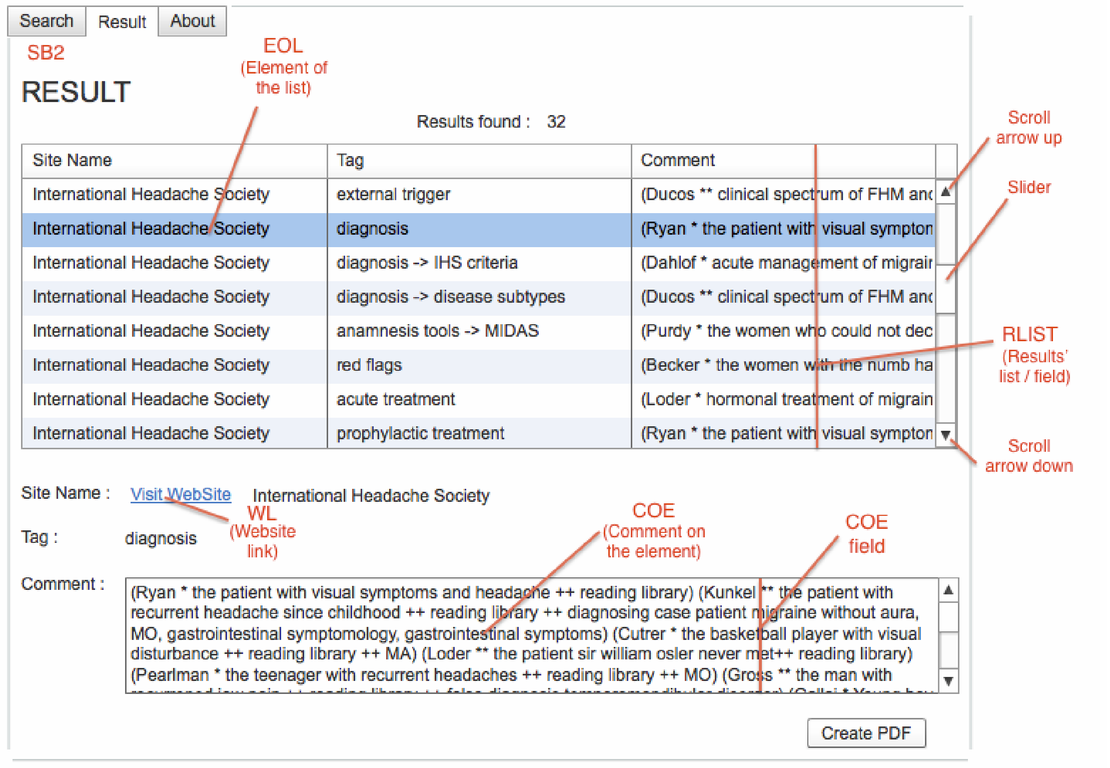

FIGURE 4.2: Search engine, results' page perspective

results, including a website name, associated website tag, and a short comment, related to the chosen website. Depending on a number of displayed results, the scrollbar may appear, if the result's list exceeds the size of the list box. The list box may display eight elements at most, where the rest is accessible through the scrollbar.

The second interface element is a message box, which displays the full comments, corresponding to the currently selected result's element in the list box. Similarly to the previous case of a list box, if the body text of comments exceeds the size of the message box, the scrollbar appears accordingly on the right side of the interface element. Additionally, the user may access the related website, which is associated with the chosen result element, and is displayed at the moment of a result's selection. The user may always navigate backward from the given perspective to the perspective of the search page by clicking the upper-left button of *Search*. Conjointly, both interface perspectives allow the user, on the one hand, to invoke a search query and, on the other hand, to perform the results' analysis by simple navigation within the XML database. It is important to emphasise that navigation within the database is done by means of manipulating the *detectable* user interface elements.

As we may notice, a chosen type of the search engine can be characterised as fundamental, in relation to its functional characteristics and provided service quality. From a programming point of view, the given search engine's characteristics represent consequently a suitable testing solution, allowing to implement and to evaluate the probabilistic principles of our proactive module. In other words, our main objective is to use the fundamental search engine's structure for the purposes of amplifying its basic functions with more intricate concepts of a probabilistic data inference and proactive anticipation of a user's search intent.

4.1.2 Amplification of the search engine's basic functions

The search engine's initial functions and features can be characterised as elementary in comparison and relation to more complex computational methodologies. Similarly to the Moodle environment, the characteristics of the given target framework give us the necessary and relatively flexible *action-space*, allowing to implement and to test new computational approaches and techniques. The *action-space* represents a vital strategic choice, characterising our methodological objectives of the proactive system implementation. In other words, the given strategy represents our deliberate intention to use the search engine framework for the purposes of testing a new, not yet fully developed experimental concept.

Therefore, the combination of target-system characteristics, providing the flexible action-space, and our initial methodological objectives, represent one of the justifying motives for choosing a search engine as the testing ground for our proactive module. The underlying principles of the implemented and tested features have to reflect the probabilistic part of a system's proactive behaviour, and thus to fulfil our second research objective to validate the conjoint part of the proactive framework approach. An extension and elaboration of search engine's basic functions involve furthermore a process of new features' design, their successive implementation and testing.

Our main research objective in the given exploratory study is to enhance the elementary search engine's functions beyond the boundaries of its existing features. The short literature review of the currently prevailing methodologies of the SE domain, presented above, contains many theoretical directions, which aim to elaborate the basic search engines' functions towards more complex computational paradigms. Notably, the aforementioned directions of amplifying the SE environment include such approaches as modelling of the user's search patterns, analysis of user's browsing behaviour, modelling of user's search activity and so on.

We use the given theoretical orientations as the main coordinating factor of our study. This consequently allows us to elaborate our own methodology, which uses the aforementioned theoretical orientations but nevertheless involves more than just a simple analysis of a user's search behaviour. In order to amplify the search engine, we have to go beyond the limits of its basic functionality. This implies that we have to take into consideration more complex design approaches than just a basic analysis of user's activity logs.

First, in our research, we do not simply resort to an analysis of decontextualised user's activity logs, and thus provide suggestions, but instead, we try to relate the selected patterns of a user's search activity to the associated instances of his/her manifested cognitive states. Consequently, after correlating the user's specific patterns of a search behaviour, we try to estimate, through the probabilistic calculations, the user's currently prevailing cognitive states, and thus to anticipate the undesirable context characteristics of low relevant results. The key objectives of the proactive anticipation, in the given case, are mainly characterised by the probabilistic inference of user's mental variations, represented respectively either by a cognitive state of a user's satisfaction or dissatisfaction.

Data correlation. The activity logs, highlighting the user's search behaviour, if analysed and interpreted separately, represent only a decontextualised data, which does not provide an extensive and complete picture about the user's search intent and his/her prevailing mental characteristics. Therefore, the user's activity logs have to be analysed in conjunction with a thoroughly designed methodology, which will allow us to emphasise the log's value, and thus to benefit from its confining data. The user's prevailing mental states represent an abstract-level reflection of his/her overtly manifested physical actions, performed within the domain of the search engine environment.

Consequently, in order to fulfil our first mentioned objective of the data correlation, we have to elaborate a comprehensive methodology, which will allow us to associate the chosen cognitive states of satisfaction and dissatisfaction with the corresponding, overtly manifested instances of a user's search behaviour. In order to fulfil the aforementioned objectives, we have to resort to the cognitive modelling methodology, which will allow us to design the associated instances of a user's behaviour by means of its gradual fragmentation into smaller patterns of the overtly manifested, search-related user's actions. Concurrently, the given approach allows us to directly associate all individual mini-instances of a user's behaviour with their corresponding abstract meanings and cognitive representations.

Our modelling methodology consists of several successive steps, including:

1. Specification of interesting cognitive phenomena,
2. Visualisation of the chosen phenomena in form of their outer manifestations within the SE environment,

3. Fragmentation of the phenomena's physical representations into smaller patterns, and ultimately
4. Correlation of the selected mini-patterns with the search engines' user interface elements.

The given methodology comprises the modelling of two aforementioned cognitive phenomena, the user's cognitive state of satisfaction and dissatisfaction. The chosen approach provides us with two computational models of user's mental variations, which are ultimately implemented on the algorithmic level within the framework of our proactive system.

Data evaluation. According to the aforementioned methodological specifications, our second objective consists of integrating the mechanisms of the probabilistic evaluation, which are set to estimate the ratio of a user's currently prevailing cognitive state. We set our algorithm to evaluate, by means of the Bayesian parameter estimation, the most frequent type of a user's behavioural instance. We build our probabilistic mechanism on a basis of the accordingly parametrised statistical model, which has the ultimate objective to measure the prevalence ratio between two computational models of the user's satisfaction and dissatisfaction states. The previously correlated and assembled models, representing the user's mental variations, are subsequently used as the proactive system's main sensors for the context monitoring. Every time an instance of a particular model is detected, the corresponding data parameter is further parsed as an argument for the Bayesian model update.

The aforesaid approach allows the algorithm to detect in time a user's prevailing mental state, and thus to provide ahead the proactive system with a valuable information, used for the purposes of a context mediation. The system tries to anticipate the user's undesirable state of dissatisfaction, by progressively catching the first precondition indicators, allowing to detect an actual state of the mental dissatisfaction. The main objective in the given approach is to be able to detect the potential dissatisfaction state before a user himself has fully acknowledged the undesirable direction of an unfolding context situation. The mentioned approach characteristics provide the system with a valuable time window, allowing to initiate the necessary actions of a proactive context mediation. The detected instances of the satisfaction model provide the system with valuable data characteristics, allowing to identify in a background and to collect ahead all high relevant search results.

From a theoretical point of view, the chosen approach of providing the search engine with various proactive features, will naturally change its original conceptual characteristics and will enhance its basic functional orientations. The given conceptual shift will inherently contribute to the upgrade of the SE fundamental functionalities, and thus will move its methodological framework towards a dimension of more complex computing paradigms. One of the distinctive characteristics of such conceptual shift can be characterised by the system's acquired capabilities of the semi-autonomous functioning, which is primarily defined by the system's potential to operate without explicit instructions from a user.

In reference to the Tennenhouse's premise of the proactive computing paradigm (see section 2.1.1), the aforesaid conceptual orientation implies that our search engine can operate on a higher frequency level, without waiting for the human feedback of search criteria modifications. Furthermore, the highlighted methodological characteristics allow the search engine's operational framework to shift from the basic interactive orientation towards a more advanced proactive computing paradigm. As might be expected, in order to fully realise the aforementioned objective, the target SE environment has to be enhanced more that by few simple proactive scenarios. All system's further elaborations, requiring additional research efforts, are considered to be beyond the scope of the current PhD thesis and are ascribed as guiding objectives to the future work. However, an experimental realisation of the given objective on the present level, may serve as a stable proof of concept and the guiding schema for all future design of the proactive search engine's assistants.

The central mechanism, providing the system with a proactive behaviour is based on the design and development of corresponding proactive scenarios. Every scenario is programmed to take care of a specific context setting. We must build the related types of a system's proactive behaviour, basing on the design and development of the SE-specific proactive scenarios. Our objective is therefore to design the proactive scenarios, which on the one hand, amplify the search engine's basic functions and, on the other hand, reflect the needs and objectives of a user's search querying activity.

We may logically assume that the instances of the mentioned user's activity may include a multitude of corresponding behavioural patterns, and thus to cause the substantial context variations. However, according to the current study objectives, we focus on the design and conception of proactive rules, which only cover the most important aspects of a user's querying behaviour. Consequently, through the implementation and testing, the given proactive rules will help us to validate and to empirically ground open theoretical concepts and approaches. In the upcoming sections we present in greater details the SE-specific proactive mechanism, where we describe the types and variations of all search engine's proactive scenarios and rules.

4.1.3 Probabilistic attributes of the proactive search engine

According to the guiding schema of our methodology, the main objective of the given exploratory study is to re-apply the concept of proactivity within a new framework of the probabilistic approach. The realisation of the mentioned objective is mainly based on a process of re-using the previously formalised deterministic framework as the underlying basis for system's new functional principles. The given research phase represents an important methodological step, playing the key role in an overall evolution of the proactive system's approach. The application of probabilistic principles allows us to implement more flexible and dynamic mechanisms of a context monitoring, event evaluation and data matching.

In comparison to the earlier discussed deterministic principles, which aimed to integrate the concept of proactivity on the fundamental uniform level, the successive probabilistic framework will consist of slightly different computational techniques and methods. However, if the new means and techniques are different at a certain level, they all are based on, and build upon our previously formalised deterministic framework with the same underlying objective of investigating the proactive computing paradigm.

In addition to a general approach, which characterises the operation of proactive scenarios, we borrow from the prior-phase methodology a deterministic mechanism of the context data detection, used within the structure of a proactive rule. All SE-based proactive scenarios are designed along the principles of the uniform function execution. In other words, for the purposes of a target context monitoring, the algorithm uses the predefined functions, which aim to detect only the designated types of data instances, representing accordingly the specified occurrences of context events. The initial experiment's objectives, which stipulate that the system must identify within a context environment the exact patterns of a user's behaviour, represent consequently our main justifying reason of employing the given methodology.

The aforesaid approach allows the algorithm to approximate the acquired data towards one of our cognitive models, and thus to identify the user's prevailing cognitive state. Only in the given manner, we are able to accurately identify an according overt manifestation of user's mental characteristics. The deterministic approach plays an important role in our algorithm, as it allows us to identify all corresponding data instances, required for the subsequent probabilistic evaluations. The chosen approach is applied upon all new SE-based proactive scenarios, which have the objective to identify within a target environment only the predefined instances of a user's behaviour. Additionally, all proactive scenarios are build upon, and based on the earlier established rule's template, which fully governs the execution of scenarios' composite functions. This represents an important methodological basis, which allows us to significantly extend the system's initial architecture by building upon a more complex framework of the probabilistic approach.

According to the conceptual characteristics of the models' design, every detected instance of a user's search behaviour, belongs either to the model of satisfaction or dissatisfaction. Consequently, the function of the statistical analysis is set to evaluate the relevance of currently displayed results, basing on a type of the received data, which are accordingly provided by our two models. The data evaluation is always taking place as a parallel process, in relation to the activity of running proactive scenarios. The data analysis itself is based on the principles of the Bayesian parameter estimation, where the Bayes' rule is set to iteratively update the values of its prior probabilities with respect to a type of the detected event. Every detected instance of a user's behaviour is communicated by the algorithm to the statistical module for its further evaluation. The detected instance of a user's behaviour aims to serve as an input data for the Bayes' rule, which allows the algorithm to calculate the proportional relations between two cognitive models and to identify the relevance level of the current results.

The given approach suggests, if the Bayes' rule evaluates the currently displayed results as high relevant, a model, which reflects the user's satisfaction state, is therefore currently prevailing. The same is valid for the dissatisfaction model. If the opposite type of the model prevails, or in other words, if the predominant part of the detected user's behavioural patterns are from dissatisfaction model, the Bayes' rule will evaluate the currently displayed results as low relevant. Consequently, during the evaluation process, the module of the statistical analysis always has its outcome values in form of a specific ratio of high and low relevant results. The given output data is subsequently used by the next-step processes of the proactive system functions.

The next third step is characterised by the system's internal monitoring activity, which represents a new type of a proactive process. Similarly to the earlier discussed principles of the autonomic computing, in the present case, the dedicated algorithm will start to monitor the proactive system's internal processes, related to the module's activity of the statistical analysis. For the given purposes, we introduce a new type of the proactive scenario, entitled in our study as *Model monitoring Meta-scenario (MTA-m).* The scenario aims to create a link between the module of the statistical analysis and a triggering mechanism of Target-scenarios.

The MTA-m proactive scenario represents the main model monitoring mechanism, allowing to detect the currently prevailing cognitive model by the continuous comparison of statistical outcome values with the predefined model's thresholds. As we will see later in greater details, the thresholds of specific values mark the absolute endmost moments of triggering the corresponding proactive actions. The MTA-m scenario has always to be aware about the values' distribution, reflecting the prevalence ratio between two cognitive models. We may notice that the conceptual design of the current triggering mechanism varies considerably in its structure in comparison to the previously discussed triggering mechanism of the deterministic framework. Instead of a regular MTA-scenario, which usually triggers the target-related processes, in the current experiment we decentralise the power of the Meta-scenario by introducing the specially dedicated *Model monitoring scenario*, which now fully assumes the decision-making of the triggering process.

All highlighted above mechanisms of the data collection and data inference are entirely based on a manipulation of gathered statistical data, representing a user's search activity. Consequently, statistics represents our main connecting instrument, allowing to convert the user's inner cognitive states into a computational form of two cognitive models. Both cognitive models respectively express the states of a user's emotional satisfaction and dissatisfaction. The activity of both cognitive models allows us to translate an abstract data of the user's mental variations into an algorithmic level of the proactive system's functions. In order to accomplish the aforementioned objectives, we have to analyse first all available types of the statistical data, which consequently represent the possible computational values for our models' algorithms.

We have to select first all needed and interesting types of the statistics, which express various characteristics of a user's search activity. The given step is the necessary prerequisite condition, as it allows us to allocate the chosen data into coherent computational patterns, representing the user's browsing behaviour. Allocated in a predefined manner, the given behavioural patterns collectively represent a specific cognitive model of one of the user's manifested mental variations. Consequently, it is of high importance to choose only the necessary, descriptive types of statistics, which will allow us to connect our models' algorithms to an abstract context domain. For an effective modelling of the user's cognitive states, we have to resort to the methodological repository and theoretical expertise of the applied cognitive psychology.

4.2 Simulation of the user's cognitive states

In order to properly allocate the collected statistical data, we need to employ a set of assisting mechanisms, which will allow us to effectively simulate the chosen variations of the user's cognitive states. The given tools will help us to define all composite elements of satisfaction and dissatisfaction states in relation to the user's cognitive processes and expressed overt manifestations. In this respect, the *Cognitive psychology* assists us in visualising the user's mental mechanisms of the information processing, which initially characterise all types of user's behavioural manifestations [55].

This leads us to the next step of studying and modelling the associated attributes of the interconnected human cognitive processes, including the human attention, memory and mental associations. The modelling itself is based on the theoretical assumptions, supported by the expertise of the applied cognitive psychology. Specifically, upon the modelling process we try to identify certain variations of the user's cognitive processes, which can be manifested by a user during an SE-based activity. Giving the specific characteristics of the search engine's user interface, we try to visualise how the user's cognitive states of satisfaction and dissatisfaction may be manifested. The hypothesis models are therefore conditioned by, and related only to the current type of the search engine's environment.

4.2.1 Underlying principles of the cognitive modelling approach

It should be noted that we do not conduct an extensive, all-inclusive study on how the human cognition works, as such study clearly lies outside of the scope of the current research project. Our main objective is to scientifically rationalise the proactive system design with regard to the underlying aspects of the human cognitive and behavioural variations, expressed within the context environment. In our study we simply highlight the cognitive-stipulated factors of a user's browsing activity, which ultimately constitute a minimal basis, required for the proactive system's operation on a proof-of-concept level. Therefore, our hypothesis models of the user's cognitive variations mainly consist of the assumption-based manipulations of only three cognitive elements, attention, memory and mental associations.

Attention. The cognitive attribute of attention, plays an important role throughout the entire design process of our cognitive models. Given the type of the chosen environment, attention represents the user's first cognitive instrument, allowing through the visual stimuli to connect the meanings of outside objects (search query, results, elements of GUI) to the user's existing repository of knowledge (memory). Throughout the search activity the user intentionally and voluntarily coordinates and allocates his attention in accordance with the characteristics and meanings of the currently displayed user interface elements. The user's applied type of attention may be correspondingly defined as the goal-driven type [86]. The *goal-driven attention* can be characterised by a set of the user's voluntarily actions, expressed in the conscious and intentional manner by means of a directed attention and visual sensory perception in relation to a particular part of the user interface.

Therefore, in our study, we use the attribute of the active attention as a valuable cognitive instrument, allowing to create a link between user's actions (manifested through the manipulation of GUI elements) and user's currently prevailing cognitive states, which consequently define the patterns of a user's behaviour. In other words, the instances of the user's manifested goal-driven attention, represent our main tool, which by means of the thoroughly defined applications, may allow us to infer the user's hidden cognitive states, search intentions and objectives. Consequently, our cognitive models have to be build in accordance with the aforementioned characteristics, which allows us to observe the user's active attention through the continuous monitoring of his browsing behaviour.

The given behavioural patterns, indicate the temporary spots of the user's present attention, clustered around and manifested within the particular parts of the search engine's user interface. Consequently, by means of the thoroughly defined techniques, we can measure, in time values, the various periods of the user's manifested attention upon particular elements of the user interface. The manifested instances of attention ultimately indicate the clusters of the user's interests, elements of voluntarily concentration, attraction, non-attraction and so on.

The time values and their variations play an important role in a cognitive model design, allowing to specify the user's individual behavioural patterns, used as the compound elements of a confining model. For example, in case of the extended, "difficult search", which accordingly requires the considerable concentration efforts, the focusing attention of a user may start gradually decrease with every unmet expectation, indicating respectively the low relevant results [87]. The user's prevailing cognitive characteristics, expressed by the state of dissatisfaction, are correspondingly reflected through the user's manifested instances of *inconsistent* actions, performed upon the navigation within different elements of the user interface. The different time values, representing the spans of a user's active attention, may be effectively expressed by implemented timers of the data collecting techniques. The given time values, represent the valuable data types, which upon the thoroughly defined application may successively display the moments of a user's high concentration and the moments of respective focusing decays.

The clusters of a user's high concentration or the attention decay allow us to visualise their corresponding representations within the search engine's user interface. The identified GUI elements may reflect in real time the clusters of the user's potential interests, and thus to indicate the first prerequisites of the user's cognitive state of satisfaction. In the similar way, the clusters of the user's focusing decays, abrupt and inconsistent actions, captured by the dedicated sets of statistics, represent collectively the potential first prerequisites of the user's cognitive state of dissatisfaction. It should be noted that upon the modelling of the user's behaviour, we have to define quantitatively what it means for a data to be extensive or non-extensive, large or small and so on.

The user's cognitive attribute of the goal-driven attention plays the role of a vital connector between the user's manifested actions, employed upon objects of interests (search results), and the user's initial knowledge about these objects, stored in the network of his/her mental memory. The memory and a network of its mental associations represent therefore an important compound aspect in our attempt to model the variations of the user's cognitive states.

Memory. The user's cognitive attribute of memory is our second important element, which is taken into consideration during the design of cognitive models. Our motivation to employ the attribute of attention has been stipulated by the fact that it allows us create a link between the physical domain of user's actions and abstract domain of user's mental representations and knowledge. The memory, symbolising the user's mental repository of knowledge about physical phenomena and their interrelations, represents our second cognitive attribute, which allows us through the methodical observations of user's actions to infer his/her potential positions, feelings and general mental attitudes towards currently prevailing context conditions. In our study, we resort to functions of the specific cognitive component, where the information association and information manipulation is performed on a short-term basis. Thus, we take into consideration the user's *Working memory*, which is defined as a memory system of the limited capacity, allowing to temporarily store and to manipulate the primary information, involved in the performance of ongoing cognitive tasks [88].

The primary reason for choosing the given type of a memory system is due to its characteristics of operating only in relation to the currently prevailing cognitive tasks. In our study, we uniquely rely on user's short-term cognitive mechanisms, which restrictively operate only in relation to the currently unfolding context conditions. Such position allows us to ensure the link between user's actions, his/her currently prevailing mental characteristics and the active conditions of the present context settings. Consequently, by taking into consideration the user's working memory, we know that the given cognitive attribute has a limited capacity, capable to simultaneously sustain only 3-4 visual objects, which are involved in the ongoing cognitive task processing [89].

Therefore, during the model design of the selected cognitive states, we try to emulate the patterns of user's potential actions, in accordance with the aforementioned disposition of the memory's short-term capacity. In other words, we try to *model* (a) the user's behavioural patterns, which reflect (b) the user's cognitive characteristics, related to a processing of the ongoing cognitive tasks, conditioned by (c) the currently present context settings. In consequence, our two models aim to *simulate* (b) the user's cognitive states, which are manifested by (a) the distinct patterns of user's actions upon the presence of (c) the specific context conditions, such as a particular category of displayed results, the identity of last viewed results and so on. Subsequently, every type of the inspected results evokes (b) an immediate information processing in form of the ongoing cognitive activity within the user's working memory system. Ultimately, the resulted output of (b) the cognitive task processing, is overtly manifested (a) in form of the user's actions, expressed as navigations within the search engine's user interface, including mouse clicks, sidebar scrolling, mouse hovering, and so on.

According to its structure, the working memory allows a user to temporarily store and to manipulate the visual images, or a verbal information, selected by the attention and captured by the corresponding sensory system. During the phase of an information processing, the interaction between memory and attention is fundamental, as the function of attention is necessary for various processing tasks of the working memory. More specifically, the visual working memory allows an external visual information to be sustained without a sensory input for a certain period of time, allowing to maintain the attention, covering the corresponding visual objects for the purposes of an ongoing cognitive task processing [90]. Additionally, regarding the relationships between the spatial visual attention and working memory, the function of attention serves not only as a mechanism of storing an information into the memory, but furthermore, it serves as a mechanism of an information retrieval from the visual working memory [91]. Consequently, the given aspect plays an important role in the process of the models' design, as it allows us to theoretically rationalise an idea that the overtly manifested actions are the immediate resulting outputs of a cognitive task processing of the working memory.

For the purposes of a more detailed decomposition of the working memory we will briefly review the Baddeley's memory model. The multi-component model of the working memory, proposed by Alan Baddeley and Graham Hitch, initially incorporates several composite modules, including the components, dealing with a phonological and spatial visual information [88]. Additionally, the model contains the *Central executive* module, which acts as a supervisory system, and thus, is responsible for the coordination of ongoing cognitive processes [92]. In other words, the given module of the working memory, represents an attentional control system, allowing to regulate the flow of information and to supervise the ongoing cognitive processes and their successive resulting outputs.

All aforementioned cognitive attributes of attention and memory, together with their underlying functional characteristics, play an important role in an overall process of the cognitive models' design. More specifically, (1) we build our simulation models, basing on the underlying mechanisms of the user's cognitive processes, which conventionally are based on the bidirectional information exchange between various cognitive attributes. (2) We make the assumptions of the user's possible context actions, which ultimately have to reflect one of the associated cognitive states. (3) Ultimately, an effective simulation of one of the cognitive models, allows us to identify the corresponding set of context conditions, including a general category of displayed results, identity of last viewed results, characteristics of individual selected results and so on. Therefore, on the basis of the aforementioned three steps of (1) analysing and planning, (2) postulating and implementing, and ultimately (3) model simulating, we are able to create a link between (a) user's context actions, (b) his/her defining cognitive characteristics, and (c) the corresponding context conditions, represented in form of high and low relevant results.

4.2.2 Characteristics of the users' browsing behaviour

From an empirical point of view, in order to connect the user's context actions to their corresponding abstract meanings, we have to identify first the specific patterns of a user's behaviour, which reflect the instances of the corresponding states of satisfaction and dissatisfaction. The given specific patterns may have different forms and variations, depending on a cognitive state they aim to represent. However, all types of the user's behavioural patterns are essentially based on the specific manipulations of corresponding user interface elements and their combinations with various timers and mouse hovering activities. For instance, a pattern may consist of a series of clicks upon the results, allocated within a particular time frame, followed by an action of scrolling down the sidebar; or a pattern may consist of two specific clicks, one upon a result, the other upon the corresponding website link, accompanied by a mouse hovering activity in a specific zone of the user interface. Every element of the search engine's user interface in itself contains a specific abstract meaning, in relation to its function or its content. Collectively, the specific combinations of user interface elements, in conjunction with the predefined timers and other types of implemented statistics may at certain level represent a particular abstract meaning, in relation to an emulated type of the user's cognitive characteristics.

During the process of modelling the variations of user's cognitive states, first, we take into consideration the potential abstract meanings of certain context actions, which may be derived from various combinations of the user interface elements. It should be noted that the corresponding examples of the given statistical combinations, presented below, simply aim to demonstrate the character of the taken first-step procedures, pertinent to the specifications of a cognitive modelling process. Therefore, all presented illustrations are relative, which consequently implies that the assigned abstract meanings may in certain context conditions be accordingly converted to their opposite values.

Our first illustration is related to the **time variations between various types of user's actions**. For instance a time variation between a mouse click on the results' element and a mouse click on the corresponding website link may successively indicate the different types of user's mental dispositions, related to his/her initial objective to obtain the desired results. A very short time interval may potentially indicate the user's high interest, and a considerable importance of the selected result. A large time interval, in contrast, may indicate the user's uncertainty in his/her search intentions, and the low relevance of the selected result. In certain context conditions, a large time interval may reversely indicate the user's diligence in choosing the accurate results.

The next context interpretation addresses the issue of **reviewing an element of the search results**. In the given case, the reviewing a result, which has been previously clicked, may potentially indicate the user's willingness to return to the inspected part of the results' list due to his/her uncertainty in a general category of the displayed results. However, the selected result, in the given context conditions, is presumed to be the closest option of choice to the user's general interests. Consequently, all reviewed elements may be considered as clusters of interests, and thus marked accordingly.

Skimmed viewing of the results, characterised by actions of scrolling the sidebar without interacting with the results elements and skipping the non-viewed items, may potentially indicate the overall low relevance of the displayed results. Additionally, the given browsing characteristics may demonstrate the user's chaotic approach to a search, which consequently may be influenced by the user's low interest in the given results' category, or by his/her dissatisfaction, related to the previously viewed results. Furthermore, upon certain context conditions, the given behavioural characteristics may reversely indicate the opposite values of their abstract meanings. The skimmed viewing may, in the given case, be considered as the targeted results' browsing, which is consequently based on the user's very specific search objectives.

Upon a first-time display of the results, a **scrolling of the results' list** down to the bottom of the results' page, then scrolling it backwards all way up to the beginning of the list, may potentially indicate the user's willingness to take an overview of all results. In certain context conditions, where the user for example navigates away from the list, the given behavioural pattern may consequently suggest the presence of low relevant results, which do not satisfy the user's current search objectives.

The next context interpretation addresses the issue of a **nonconsecutive selection**. Upon the nonconsecutive selection of results' elements, all skipped and subsequently viewed elements may eventually be considered as the markers of the user's growing interest in relation to the given category of viewed results. Subsequently, the given results' elements can be marked as middle or high relevant for the successive algorithm's calculations, which use the given markers as the function parameters for a background search of the similar or alternative results.

The stopping points or locations of a scrollbar may eventually suggest the clusters of user's attentional shifts, which correspondingly indicate the locations or items of user's potential interests, represented visually in form of results or comments.

The last category of a context activity, used in the design of our cognitive models are the instances of a user's **mouse hovering activity**, performed over the different elements and zones of the user interface. Upon the condition that a mouse hovering activity is consistent, the given context action may successively indicate the direction and accentuation of the user's attention in relation to a specific zone of the search engine's user interface. Consequently, the content, which is currently displayed under the position of the cursor may indicate a type and category of the user's potential interests. In certain context conditions, the extended hovering time over an element of the user interface may successively indicate the user's state of uncertainty and reflection; or in contrast, it may suggest the importance of an accurate choice. The mouse hovering and movement of the cursor throughout the length of a sentence or a phrase may suggest the user's accentuated and focused attention in relation to the given element. Subsequently, the mentioned type of the focused attention indicates the user's willingness either to accurately understand the meaning of what is indicated or to confirm his/her initial search objectives with the characteristics of the currently displayed information. We may successively presume that the characteristics of the currently displayed results' element, represent the closest option of choice to the user's general interests and search objectives.

The mouse hovering in itself, plays an important role in a deducting process of the user's cognitive states. Conventionally, in our statistical data collection, we monitor the mouse hovering activity in two different zones of the search engine's user interface. One is related to the RLIST field (mhR), another is related to the COE field (mhC) (see Figure 4.2). Below, we present the detailed examples, which aim to illustrate the application of the user's specific mouse hovering activity within RLIST and COE fields. The following example is build on a basis of our assumptions, related to the user's possible behavioural patterns, manifested upon viewing of the displayed search results. Due to the search engine's capability of presenting eight results' elements simultaneously, we presume that at the moment when a user sees for the first time the generated list of results, there may be a short pause in a user's mouse hovering activity between the moment SB1 is pushed and the results are displayed. We hypothesise that such situation may be true, given the fact that the user may spend a short period of time for the purposes of scanning the recently displayed results' list.

The next possible step is an interaction with the list, including the mouse hovering over the results' list, followed by the successive interaction with the individual results' elements. At the moment of selecting an element in the list, there is a possibility that the position of the hovering cursor will stay unchanged for a few seconds, as at this moment the comments, which are associated with the selected element are displayed. Upon the display of comments, the user's attention is immediately shifted towards the corresponding section of the COE field with the few seconds delay of the following motor function of the cursor repositioning. At a certain moment, depending on a relevance level of the selected result and focusing level of the user's active attention, the location of the cursor may shift towards an area of the displayed comments. Additionally, the implemented timers, which are activated upon the beginning of a user's interaction with the results' list may help us to associate and to relate an mhR data to the specific time frame, and thus to identify the duration of a mouse hovering activity.

Regarding the example of the mouse hovering activity within COE field, we presume that there may be a short pause in the user's cursor movement between the period of selecting the results' element and viewing the displayed results' comments, as user's attention shifts first towards the COE field and only then it is followed by the motor function of a mouse movement. Upon the selection of the results' element, few lines are displayed in the field of comments, which implies that if the given result is of high relevance, the user may hover over the lines while reading, or even try to select a certain text. Such actions are the primary indicators of the accentuation of a user's attention, which may reveal the rise of a user's interest in the displayed results. Similarly to the previous example, the implemented timers, which are activated upon the beginning of a user's interaction with the comments may help us to associate and to relate an mhC data to a specific time frame, and thus to identify the duration of a mouse hovering activity. The specific correlations between the mouse activity and duration of viewing the comments may indicate the fall or rise of a user's interest in relation to the selected search result.

The presented above examples, illustrating an interpretation of the user's possible context actions and successive assignment of corresponding abstract meanings, have the objective to demonstrate from a general point of view the preliminary steps of cognitive modelling mechanisms. All abstract meanings, which have been assigned to a combination of user's actions, postulated on the basis of the presented earlier theoretical framework of the cognitive information processing, ultimately aim to represent, in a collective form, a complete model of one of the selected cognitive phenomena. In our upcoming sections, we demonstrate the mechanisms of the cognitive modelling methodology, which allows us, in a similar manner, to model the patterns of user's context actions, simulating respectively the user's two cognitive states. Conventionally, this is the mental state of a user's satisfaction in relation to the high relevant results and the state of a user's dissatisfaction in relation to the low relevant results.

The simulation of user's cognitive states is mainly performed by means of the running models, which initially enclose the various modelled patterns of a user's overt behaviour. Therefore, the initial objective of a cognitive modelling approach is to simulate the variations of user's cognitive processes on a computational level [58]. In order to achieve the aforementioned objective, we use statistics as the main technique, allowing the proactive system's algorithms to create a link between user's actions and their corresponding abstract representations. By using the chosen statistics as the context-related input data, which typifies the instances of a user's overt behaviour on an algorithmic level, we are able to define the specifics of user's mental variations. Consequently, a particular instance of a statistical data, allocated into a static precise sequence, successively represents a corresponding aspect of one of the user's cognitive states. The main approach presumes the allocation and exact arrangement of a statistical data into the logical and coherent sequences, chain of sequences, patterns and eventually models. In the end, the complete model must be able to fully represent a particular cognitive state, manifested by a user upon his/her browsing activity.

Ultimately, each model will consist of an equal number of the composite statistical sequences or mini patterns, which collectively represent a full model. The given approach, allows us to perform an accurate Bayesian parameter estimation of a user's cognitive state prevalence, with equal possibilities for each model. Every type of a cognitive model essentially contains ten mini patterns, representing the individual instances of a user's browsing behaviour, which are successively characterised as input data types. If a mini pattern is detected, the corresponding function will successively return a related value, which will ultimately serve as the input data for the Bayes' formula.

Conventionally, we have ten composite mini patterns of the dissatisfaction model, included in, and represented by the special *Context monitoring Meta-scenario (MTA-c)*, which monitors the user's context actions for the purposes of detecting the instances of the *browsing inconsistency* and possible manifestations of the *emotional discontent*. Additionally, we have ten composite mini patterns of the satisfactions model, which are equally represented by the same Meta-scenario (MTA-c), which furthermore monitors the user's context actions for the purposes of detecting the instances of the *browsing consistency* and possible manifestations of the *emotional contentment*. Ultimately, each mini pattern represents a short sequence of specific statistical data, allocated in a way, which allows us to identify the various aspects of user's prevailing cognitive characteristics. The data, which represents the user's contentment and consistency, is subsequently used by an algorithm as a benchmark or criterion for a search of the similar and alternative results.

4.2.3 Statistical components of cognitive models

The use of statistics in our study represents the algorithm's prime technique to connect and to relate the low level data to the specifics of user's cognitive characteristics. If collected and allocated properly, the statistics may reveal various types of information, which ultimately specifies the user's contextual characteristics, including his/her cognitive tendencies, search objectives and interests. In order to achieve the aforementioned objective, we employ several types of statistics, including time detectors, timers, selection data, mouse clicks, mouse hovering, time intervals, data interrelations, data associations and disassociations. The various combinations of the given statistical data, ultimately allow us to perform the algorithmic computations upon the detected data for the purposes of revealing the user's hidden cognitive states, disguised under different settings of context conditions. In order to allow the system to perform the statistical data collection, we elaborate and implement a mechanism, consisting of several methodological steps.

First, we define the related types of statistics, collected from a user's browsing activity, with the objective to reveal the various aspects of user's prevailing mental characteristics. Upon the second step, according to a type of a cognitive model, we try to identify the different statistical relationships, data correlations and disassociations, which in their aggregated form aim to represent the specific instance of a user's emotional expression. As we mentioned previously, a sequence of the predefined and allocated statistics, constitutes a complete cognitive model. Therefore, in order to detect a model, the associated composite algorithms, included into a structure of the context monitoring scenario (MTA-c), must progressively detect or to reveal from an empirical data, all model's compound individual patterns, representing the user's propagating actions. For the given purposes, we design several dedicated proactive rules, included into the structure of the MTA-c, where each rule has its own objective to detect only the specific pattern of a user's search behaviour. Consequently, an entire model, that is a collection of ten specific statistical combinations, is detected by the given ten proactive rules of the MTA-c scenario, which is initially set to continuously monitor the search engine's current status.

For the purposes of the given experiments, we decide to collect the statistical data, reflecting various instances of a user's behaviour from both, the search page and results' page perspectives. However, for the specific task of detecting the user's behavioural reactions and analysing his/her emotional inclinations and mental attitudes towards the displayed results, we implement the monitoring algorithms only in relation to the results' page perspective. The statistical data analysis of the search page perspective is left for a potential consideration of a future research.

Results' page perspective. In relation to the perspective of the results' page, the underlying objective of implemented algorithms is to detect the user's prevailing behavioural patterns, reflecting either a cognitive state of satisfaction or dissatisfaction. A variety of the collected data represents therefore a specific pattern type, allocated within one of the cognitive models. For the purposes of detecting both model types, we have to respectively implement within the results' page perspective ten monitoring algorithms of the satisfaction model and ten monitoring algorithms of the dissatisfaction model. Consequently, we have twenty dedicated proactive rules of the MTA-c scenario, which are set to continuously observe the results' page perspective for the specified patterns of a user's behaviour.

According to the aforementioned characteristics, we successively specify the exact elements of the search engine's user interface, which will be involved in a data detection. It should be noted that all statistical data is classified into several categories, including mouse clicks, timers, mouse hovering and sidebar scrolling. In Tables 4.1, 4.2 and 4.3 we present the statistics' summary, arranged into the corresponding categories, which aim to highlight the general characteristics of all collected data.

In Table 4.1, we may notice the presence of the SB1 element, which is the only GUI element from the search page perspective. Moreover, the user's successive interactions with the Search Button SB1, allow the corresponding algorithm to start the timer#6, which is used by certain proactive rules as a general time reference. The given timer is set to run continuously during a period of the user's interactions with the results' page until the Search Button SB2 is clicked. The given approach allows the proactive system to associate the different types of user's actions with the given time reference, and thus to deduct and to specify the points of the beginning and the end of an action, as well as a length of a specific user's activity. If needed, we can compare the values of other timers with the reference timer#6, such as in the case of EOL selection, website access, or mouse hovering.

The rest of the presented timers from Table 4.1 have the common objective to help us to place a mouse hovering data into a time relationship for the purposes of identifying

TABLE 4.1: Timers

Timers	Location on UI	Timer starts	Timer stops	Data to save	Comments
Timer #6	SB1	When clicking on SB1 with condition that current search query has results to display;	When clicking on SB2 or at the condition of page timeout;	Save the total time of timer #6;	The general time reference for result page;
Timer #7	RLIST	When the cursor enters RLIST field;	When the cursor leaves the RLIST field;	• Total time of mouse hovering over the RLIST field; • Every starting time of mhR in relation to timer #6;	mhR - mouse hovering over RLIST field;
Timer #8	COE	When the cursor enters COE field;	When the cursor leaves the COE field;	• Total time of mouse hovering over the COE field; • Every starting time of mhC in relation to timer #6;	mhC - mouse hovering over COE field;

the average duration of the user's mouse activities within the RLIST (mhR) and COE fields (mhC). The time duration, or a total time of a mouse hovering activity over the RLIST and COE fields may indicate, upon the specific context conditions, a level of the user's interest in an associated results' element. Additionally, a continuous positioning of the cursor over the respective fields of the search engine's user interface may suggest an extended accentuation of the user's focusing attention upon the hovered elements of a potential interest. The given factor may consequently indicate a user's willingness to study or to relate, by means of cognitive associations, the meta-information about the displayed results to the mental representations of a similar knowledge, stored in the user's memory. Furthermore, the starting points of timers 7 and 8, will subsequently allow us to measure a pause between user's activities in the RLIST and COE fields. The given timers will allow us to place the user's specific actions in a time relationship of the timer 6. Successively, the given approach characteristics make possible to create the statistical correlations and associations between the context data for the purposes of modelling the individual instances of a user's behaviour.

The Table 4.2 contains an information, pertinent to two separate statistical instances of the results' selection (EOLs) and a website link access (t_{WL}). The first statistical instance EOLs, aims to represent a user's context action of selecting a candidate element in the displayed list of the search results. By means of the given statistical element, we are able to detect several properties of the selected result, including its associated website name, website tag, position of en element in the global results' list and a time of an EOL selection in relation to the timer 6. Consequently, the given data types allow

Table 4.2: Various GUI elements

Abbreviation	Type of user activity	Location on UI	Data to save
EOLs	Selection of an element in the results' list	RLIST	• Website name; • Tag; • Position in the global results' list; • Time EOL has been clicked in relation to timer #6;
t_{WL}	Time of website access	WL	Time the website link has been clicked in relation to timer #6;

us to specify the necessary attributes of all selected elements, which are successively required for the purposes of further computations by various functions of the proactive system's algorithms.

The saved website names and their associate tags, upon specific context conditions, will allow a related algorithm to identify the candidate alternative solutions and the potential similar results. An EOL's position in relation to the global results' list, together with its associated selection time, will allow us to create a schematic representation of the user's browsing characteristics. The details of such characteristics will include an average time of examining the results' element, ratio between skipped and viewed results, as well as an average progression step of the skimming browsing.

The second statistical instance t_{WL}, or a time of a website access, allows us to detect and to save a time value in relation to the timer 6, which indicates the user's click upon the associated website link of the selected result. The given statistical element allows a relate algorithm to identify the time variations between an EOL selection and website access. In other words, we try to specify how much time the user has spent on the examination of the selected EOL, before accessing its associated website link. Furthermore, the given statistical instance of a time relationship, allows the algorithm to measure an average time, spent by a user outside of the search engine, after being redirected to a related EOL website. A time, which has been spent outside of the search engine may successively indicate, upon the specific context conditions, the relevance level of the selected results' element.

The Table 4.3 contains the corresponding descriptive information, pertinent to two statistical instances of a scrollbar activity, that is the detection of a time value in relation to the sidebar arrow release (t_{AR}) and the detection of a time value in relation to the sidebar slider release (t_{SR}). Both elements consequently allow us to detect and to save the corresponding time values, indicating the stopping points of a scrollbar activity in relation to the reference timer 6. The given statistical instances, by means of applying

TABLE 4.3: Scrollbar activity

Abbreviation	Type of scrollbar activity	Data to save
t_{AR}	Scroll arrow release	Time of scroll arrow release in relation to timer #6;
t_{SR}	Slider release	Time of slider release in relation to timer #6;

the time relationships, allow us to measure the duration, or an average time of the user's stay on a particular position of the results' list. Additionally, the given statistical data allows the algorithm to identify a type of the user's browsing, for instance a slow pace browsing, represented by regular and extended intervals in the sidebar activity, or a skimming browsing, represented by fast and erratic movements of the scroll arrow or the slider. However, as we will see later, the most important applications of the given statistical instances are done by means of data correlations with other instances of the collected information.

All aforementioned statistical data, separately and by itself is not capable to reveal any valuable information about the prevailing cognitive states, related to a user's browsing activity. Independently, the given data only represents the detached and unconnected pieces of a context information, which are unable to provide any important abstract meanings by themselves. The decontextualised, single pieces of a context information are applied in a monitoring process of a user's browsing activity only as the coherent, inter-associated and correlated sequences of a statistical data. It should be noted that all aforementioned instances of the collected statistics have a functional value only upon their various combinations and accordingly organised correlations of data into sets or patterns, which subsequently play a key role in a general functioning of the system's algorithms. Only upon the given conditions, we are able to identify the hidden traits of a user's browsing activity, which successively allow us to reveal the user's most prevalent cognitive variations, associated with the specific conditions of the search engine's current state.

4.2.4 Dissatisfaction model

Our first cognitive model, entitled as *Dissatisfaction model* (D-model), aims to represent on a computational level the cognitive state of the user's dissatisfaction, which is manifested in relation to the currently displayed search results. The simulation of the given

Table 4.4: Pattern D1

Type of selected statistics	EOL_n (selection of the results' element)
Main conditions	$\Delta EOL = (EOL_{n+1} - EOL_n) \geq 10EOL$
Secondary conditions	All actions are allowed between every EOL_n selection.

state is based on the detection of the user's overtly present instances of an emotional discontent, which are stipulated by the context conditions of low relevant results. The cognitive state of dissatisfaction is identified by means of detecting the instances of a user's browsing inconsistency, manifested throughout the sessions of a search querying. The dissatisfaction model is consequently based on the "indicators" of low relevant results, which have the initial objective to signal the potential presence of the user's mental state of an emotional discontent. For the purposes of achieving the aforementioned goal, we have to quantitatively define the abstract problem of a mental dissatisfaction in terms of the user's manifested context actions. Therefore, in order to represent the particular instances of a user's overt behaviour on the algorithmic level, we have to correspondingly allocate all collected statistical data into the specific computational patterns.

Consequently, for the purposes of designing the model's compound mini patterns, we elaborate a set of supporting guidelines, allowing subsequently to facilitate the process of the patterns definition. The dissatisfaction model consists of several key principles, which define the main directions of the user's pattern specifications. The given directions are essentially characterised by the user's non-extensive interactions with the results, relatively short time of results analysis, user's considerable interactions with the scrollbar and inconsistent mouse hovering activity over the COE field. The modelling process of the user's behavioural patterns is fully based on the aforementioned guidelines of the most probable types of context actions.

Collectively, all mini patterns represent a fully functional cognitive model, which is stored in a segregated form inside of the context monitoring scenario MTA-c. In total, the dissatisfaction model contains ten mini patterns of the correlated statistical data (D1–D10), represented in form of the sequential algorithmic steps. In the upcoming paragraphs, we present the model's composite patterns, by highlighting their internal algorithmic structures, corresponding abstract meanings and the character of their functions in relation to the full model. The position of the composite patterns, in relation to their number (D1–D10), does not have any special structure or logic, all patterns are enumerated with regard to a process of their gradual design.

Pattern D1. The first behavioural pattern, entitled as *D1* contains only one statistical element, that is a mouse action of the EOL selections, collected and regarded as a context-based statistical data (see Table 4.4). The pattern consists of a series of mouse clicks, executed upon the elements of the results' list (EOL selection in RLIST). The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' list for the purposes of detecting the position difference between the selected results' elements (ΔEOL). In other words, the algorithm is set to monitor the number of skipped results' elements between the EOL selections. The exact sequence of the given statistical combination is represented as:

$$\Delta EOL = (EOL_{n+1} - EOL_n) \geq 10EOL \tag{4.1}$$

Alternatively, we may delineate the user's browsing activity, representing the overall results' selections as an interval between the EOL_n and EOL_{n+1} selections:

$$[EOL_n,\ EOL_{n+1}] = \{x \,|\, EOL_n \leq x \leq EOL_{n+1},\ x \geq 10EOL\} \tag{4.2}$$

The pattern's main condition is represented by a variable x, which delineates the minimum required number of the detected EOL between the user's result selections of EOL_n and EOL_{n+1}.

The D1-pattern contains two sets of conditions, primary conditions define the process of a pattern completion, the secondary define the pattern's general flexibility. Consequently, the primary conditions specify that pattern is completed if a number of the skipped results' elements is equal to, or greater than ten EOLs ($\Delta EOL \geq 10EOL$). In other words, the algorithm marks the pattern as accomplished if the difference between the first selected results' element (EOL_n) and the next selected element (EOL_{n+1}) is equal to, or greater than ten EOLs ($10EOL$).

The second set of conditions stipulates that all types of intermediate actions, performed by a user between the EOL selections are initially allowed, including scrollbar activity, mouse hovering over the COE field, website link access and so on. It should be noted that the given second set of conditions is required for defining the level of the pattern's flexibility in relation to its final completion. In other words, by allowing the intermediate actions between the elements of a pattern, we make sure that the initial pattern will not be broken or interrupted by one of the possible user's off-pattern actions. The given schema of primary and secondary conditions is relevant to, and valid for all further presented behavioural patterns of a user.

From an abstract point of view, the D1-pattern aims to represent the user's skimming type of browsing, which may be considered as *common* or *typical* in a search querying situation, characterised by the presence of low relevant results. During a browsing

Table 4.5: Pattern D2

Type of selected statistics	EOL_n (selection of the results' element) $\Delta t_{(EOL_n)}$ (total time of stay of the selected EOL_n) $\Delta \bar{t}_{(EOL_n)}$ (average time of stay of the selected EOL_n)
Main conditions	$\Delta t_{(EOL_n)} = t_{(EOL_{n+1})} - t_{(EOL_n)}$ $\Delta \bar{t}_{(5EOL)} = \frac{1}{5} \sum_{n=1}^{5} \Delta t_{(EOL_n)} \leq 3.5sec$
Secondary conditions	All actions are allowed between every EOL_n selection.

activity, the user tends to skip more elements if the currently displayed list of search results does not meet the user's initial expectations.

Pattern D2. The second behavioural pattern, entitled as *D2* contains three statistical elements, that is a mouse action of the EOL selection, time difference or total time of stay of the selected results' element ($\Delta t_{(EOL_n)}$) and the time difference, indicating the average time of stay of the selected results' element ($\Delta \bar{t}_{(EOL_n)}$). Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.5). The pattern consists of a series of mouse clicks, executed upon the elements of the results' list (EOL selection in RLIST). The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' list for the purposes of detecting an average time of stay for every five selected results' elements.

The D2-pattern contains two sets of conditions, where primary conditions specify that the pattern is completed if an average time of stay for every five selected results' elements is equal maximum to 3.5 seconds ($\Delta \bar{t}_{(5EOL)} \leq 3.5sec$). In order to calculate the given time conditions, we must first specify the time difference, or the total time of stay for every selected results' element ($\Delta t_{(EOL_n)}$). The only time value, associated with a mouse click upon an EOL is a time value of its selection in relation to the timer 6 (see Table 4.2). Therefore, in order to acquire the time difference, or the total time of stay of the selected EOL ($\Delta t_{(EOL_n)}$) we have to deduct the time value of its selection (t_{EOL_n}) from the time value of a next selected EOL ($t_{EOL_{n+1}}$). After having specified the total time of stay for every five selected EOLs, we calculate furthermore the values of their average time of stay ($\Delta \bar{t}_{(5EOL)}$). For the given purposes we use the simple arithmetic mean equation, which is initially defined as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} a_i \tag{4.3}$$

TABLE 4.6: Pattern D3

Type of selected statistics	AR_n (sidebar arrow click)
Main conditions	$\sum AR_i \geq 15AR$
Secondary conditions	All actions are allowed between every AR_n click, except EOL_n selection, website link access and slider movements.

Consequently, in accordance with the pattern's specifications, we adapt the aforementioned equation in order to calculate the average time of stay for every five selected results' elements. Additionally, we specify the related context-based conditions, which stipulate that a calculated average time of duration for every selected EOL has to be equal maximum to 3.5 seconds:

$$\Delta\bar{t}_{(5EOL)} = \frac{1}{5}\sum_{n=1}^{5}\Delta t_{(EOL_n)} \leq 3.5sec \tag{4.4}$$

The pattern's flexibility in relation to its final completion, similarly to the D1-pattern is stipulated by the second set of the algorithm's conditions. The second set of conditions specifies that all types of intermediate actions, performed by a user between the EOL selections are initially allowed, including the scrollbar activity, mouse hovering over the COE field, website link access and so on. The D2-pattern will not be broken or interrupted if one of the given user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern D3. The given behavioural pattern, entitled as *D3* contains one statistical element, that is a mouse action of the AR clicks, collected and regarded as a context-based statistical data (see Table 4.6). The pattern consists of a series of mouse clicks, executed upon one of the sidebar arrows of the results' list. The initial objective of the given pattern is to monitor the user's interactions with various elements of the sidebar for the purposes of detecting the total number of user's mouse clicks, executed upon a sidebar arrow. The exact sequence of the given statistical combination is represented as:

$$\sum AR_i \geq 15AR \tag{4.5}$$

The D3-pattern contains two sets of conditions, where the primary conditions specify that the pattern is completed if a sum of all sidebar arrow clicks, performed by a user during a search activity is equal minimum to fifteen mouse clicks ($\sum AR_i \geq 15AR$). For the purposes of calculating the given conditions, the associated algorithm of the MTA-c

Table 4.7: Pattern D4

Type of selected statistics	AR_n (sidebar arrow click) SR_n (sidebar slider movement)
Main conditions	$\sum AR_i \geq 3AR + 1SR$
Secondary conditions	All actions are allowed between $3AR$ clicks and $1SR$ movement, except EOL_n selection and website link access.

scenario is simply set to sum up the user's mouse clicks, executed upon a sidebar arrow. In order to prevent the repetitive count of the same data, the user's mouse clicks are segregated into the sections of minimum fifteen items. As soon as one set of the fifteen arrow clicks is detected, its composite elements get successively discarded and therefore are not taken into consideration by the algorithm upon its next iteration. The given approach of the data segregation into specific sets is equally applied in all similar cases of the pattern detection.

The pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The second set of conditions specifies that all types of intermediate actions, performed by a user between the AR mouse clicks are initially allowed, excluding the EOL selection, website link access and sidebar activity trough the slider movements. It should be noted that the D3-pattern will be broken or interrupted, if one of the specified user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern D4. The given behavioural pattern, entitled as *D4* contains two statistical elements, that is a mouse action of the AR clicks and SR movements, collected and regarded as a context-based statistical data (see Table 4.7). The pattern consists of a series of mouse clicks, executed first upon a sidebar arrow and subsequently followed by a slider movement of the results' list. The initial objective of the given pattern is to monitor the user's interactions with various elements of the sidebar for the purposes of detecting an action of scrolling the results' list down through the scroll arrow with minimum 3AR clicks, followed by at least one slider movement (1SR). The exact sequence of the given statistical combination is represented as:

$$\sum AR_i \geq 3AR + 1SR \qquad (4.6)$$

The D4-pattern contains two sets of conditions, where the primary conditions specify that the pattern is completed if a sum of all sidebar arrow clicks, performed by a user

Table 4.8: Pattern D5

Type of selected statistics	WL_n (website link access) AR_n (sidebar arrow click) SR_n (sidebar slider movement) EOL_n (selection of the results' element)
Main conditions	$\left.\begin{matrix}[t_{WL}, t_{AR_1}]\\ [t_{WL}, t_{SR_1}]\\ [t_{WL}, t_{EOL_1}]\end{matrix}\right\} = \{t \,\vert\, 0\,min. \leq t \leq 2\,min.\}$
Secondary conditions	All actions are allowed between WL_n click and first following action.

during a search activity is equal minimum to three mouse clicks ($\sum AR_i \geq 3AR$), followed at least by one slider movement ($1SR$). For the purposes of detecting the given pattern, the associated algorithm of the MTA-c scenario is first set to sum up the user's arrow clicks ($3AR$). Consequently, the algorithm initiates the monitoring of the user's next action, which according to the specifications of the given pattern, is defined as the list's slider movement ($1SR$).

The pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The second set of conditions specifies that all types of intermediate actions, performed by a user between the 3AR clicks and 1SR movement are initially allowed, excluding the EOL selection and website link access. It should be noted that the D4-pattern will be broken or interrupted if one of the specified user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern D5. The given behavioural pattern, entitled as *D5* contains five statistical elements, that is a mouse action of the AR click, SR movement, EOL selection and a website access. Additionally, the pattern contains a time variable of the action duration. Collectively, all elements are regarded as a context-based statistical data (see Table 4.8). The pattern consists of a series of mouse clicks, executed first upon a website link (WL) and subsequently followed either by a sidebar arrow click (AR), slider movement of the results' list (SR), or an EOL selection.

The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting the total time of a user's stay outside of the search engine. The algorithm has to detect the time intervals between user's several context actions, including $[t_{WL}, t_{AR_1}]$, $[t_{WL}, t_{SR_1}]$ and $[t_{WL}, t_{EOL_1}]$. For the given purposes, the algorithm is first set to detect a user's mouse click upon a website link (WL), which will lead the user outside of the search engine's environment. Subsequently, the algorithm has to detect the user's very first context action within the search engine's environment, specified as the first sidebar arrow click (AR_1), first slider movement (SR_1), or the first EOL selection (EOL_1). Basing on the detected metadata, the algorithm has to deduct the time differences between given actions, which in the end will correspond to various time intervals. The exact sequences of the given statistical combinations are collectively represented as:

$$\left.\begin{array}{c}[t_{WL}, t_{AR_1}]\\ [t_{WL}, t_{SR_1}]\\ [t_{WL}, t_{EOL_1}]\end{array}\right\} = \{t \,|\, 0\,min. \leq t \leq 2\,min.\} \tag{4.7}$$

The primary conditions of the D5 statistical combinations specify that the pattern is completed, if the time intervals of the corresponding aforementioned behavioural patterns lie within the limits of $\{t \,|\, 0\,min. \leq t \leq 2\,min.\}$. For the purposes of detecting the given patterns, the associated algorithm is first set to deduct the time differences between the time values of the given user's actions, including $\Delta t_{(t_{WL}, t_{AR_1})}$, $\Delta t_{(t_{WL}, t_{SR_1})}$, and $\Delta t_{(t_{WL}, t_{EOL_1})}$. Therefore, all deducted values of time differences successively correspond to the time intervals between the user's specific actions, which according to the D5-pattern specifications have to be within the particular time limits:

$$\begin{aligned}\Delta t_{(t_{WL}, t_{AR_1})} &= [t_{WL}, t_{AR_1}] = \{t \,|\, 0\,min. \leq t \leq 2\,min.\}\\ \Delta t_{(t_{WL}, t_{SR_1})} &= [t_{WL}, t_{SR_1}] = \{t \,|\, 0\,min. \leq t \leq 2\,min.\}\\ \Delta t_{(t_{WL}, t_{EOL_1})} &= [t_{WL}, t_{EOL_1}] = \{t \,|\, 0\,min. \leq t \leq 2\,min.\}\end{aligned} \tag{4.8}$$

Consequently, the D5-pattern is marked as completed when a detected time interval of one of the three user's specified action combinations satisfies the initial time conditions of $\{t \,|\, 0\,min. \leq t \leq 2\,min.\}$.

The D5-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The second set of conditions specifies that all types of intermediate actions, performed by a user between the WL click and one of the three specified actions of either AR, SR, or EOL click are initially allowed. It should be noted that the D5-pattern will not be broken or interrupted if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern D6. The given behavioural pattern, entitled as *D6* contains two statistical elements, that is a mouse action of the EOL selection and content related metadata, which describes a type of the selected EOL by a corresponding content tag. The *uniform tag-set*, or in other words a set of two identical elements $\{a, a\}$ is defined as a selection

TABLE 4.9: Pattern D6

Type of selected statistics	EOL_n (selection of the results' element) T_n (a uniform tag-set $\{a, a\}$)
Main conditions	$\sum EOL_i \geq 3T_n$
Secondary conditions	All actions are allowed between EOL_n selections.

of the minimum two EOL with the identical descriptive tags:

$$T = \{a, a\} = \sum_{n=1}^{2} (EOL_a)_n \tag{4.9}$$

Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.9). The pattern consists of a series of mouse clicks, executed upon the elements of the results' list (EOL selection in RLIST). The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' list for the purposes of detecting minimum three different uniform tag-sets $\{a, a\}$, $\{b, b\}$, and $\{c, c\}$. In order to detect the given data, the algorithm has to continuously analyse the metadata of every selected EOL. The exact sequence of the given D6-pattern is represented as:

$$\sum EOL_i \geq 3T_n \tag{4.10}$$

Alternatively, we may delineate the user's browsing activity, representing the overall results' selections as an interval between 1^{st} and n^{th} EOL selections:

$$[EOL_1,\ EOL_n] = \{x \,|\, EOL_1 \leq x \leq EOL_n,\ x \geq 3T_n\} \tag{4.11}$$

The pattern's main condition is represented by a variable x, which delineates the minimum required number of the detected uniform tag-sets between the user's result selections of EOL_1 and EOL_n.

The D6-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The second set of conditions specifies that all types of intermediate actions, performed by a user between the EOL selections are initially allowed, including the scrollbar activity, mouse hovering over the COE field, website link access and so on. It should be noted that the D6-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Table 4.10: Pattern D7

Type of selected statistics	EOL_n (selection of the results' element) T_n (a uniform tag-set $\{a, a\}$)
Main conditions	$\sum_{i=1}^{15} EOL_i = 0T_n$
Secondary conditions	All actions are allowed between EOL_n selections.

Pattern D7. The given behavioural pattern, entitled as *D7* contains similarly two statistical elements, that is a mouse action of the EOL selection and content related metadata, which describes a type of the selected EOL by a corresponding content tag. As we defined above, the given metadata is represented by a uniform tag-set (T_n). Collectively, all pattern elements are regarded as a context-based statistical data (see Table 4.10). The pattern consists of a series of mouse clicks, executed upon the elements of the results' list (EOL selection in RLIST).

The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' list for the purposes of detecting the pattern of zero uniform tag-sets ($0T_n$) within the limit of fifteen EOL selections ($\sum_{i=1}^{15} EOL_i$). In order to detect the given data, the algorithm has to continuously take a record of every selected EOL's metadata. The exact sequence of the given D7-pattern is represented as:

$$\sum_{i=1}^{15} EOL_i = 0T_n \tag{4.12}$$

Alternatively, we may delineate the user's browsing activity, representing the results' selections as an interval between first and fifteenth EOL selections:

$$[EOL_1,\ EOL_{15}] = \{x \,|\, EOL_1 \leq x \leq EOL_{15},\ x = 0T_n\} \tag{4.13}$$

The pattern's main condition is represented by a variable x, which delineates the minimum, required number of the detected uniform tag-sets between the user's result selections of EOL_1 and EOL_{15}.

The D7-pattern contains two sets of conditions, where the primary conditions specify that the pattern is completed if a sum of fifteen selected results' elements does not contain any uniform tag-set. For the purposes of detecting the given pattern, the algorithm has the objective to monitor the sets of every fifteen EOL selections for the presence of the distinctive metadata. If no uniform tag-set has been detected, the pattern is marked as

Table 4.11: Pattern D8

Type of selected statistics	EOL_n (selection of the results' element) AR_n (sidebar arrow click) SR_n (sidebar slider movement)
Main conditions	$3(EOL_n + AR)$ $3(EOL_n + SR)$
Secondary conditions	All actions are allowed between EOL_n selection and a mouse click upon AR_n or SR_n, except a website link access (WL_n).

completed. In the opposite case, if at least one T_n has been detected, the logic of the pattern becomes broken.

The D7-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The second set of conditions specifies that all types of intermediate actions, performed by a user between the EOL selections are initially allowed, including the scrollbar activity, mouse hovering over the COE field, website link access and so on. It should be noted that the D7-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern D8. The given behavioural pattern, entitled as *D8* is characterised as a simple combination of a context data. The pattern contains three sequentially allocated statistical elements, that is a mouse action of the EOL selection, AR click and/or SR movement. Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.11).

The D8-pattern consists of a series of mouse clicks, initially executed upon the elements of the results' list and subsequently executed upon the results' list scrollbar either by means of a sidebar arrow click or a sidebar slider movement. The initial objective of the given pattern is set to monitor the user's interactions with various elements of the results' page for the purposes of detecting consecutively minimum three times the same browsing pattern, consisting respectively of one EOL selection within the results' list, followed by an action of the list scrolling by means of one sidebar arrow click or one sidebar slider movement. The pattern is marked completed, if the aforesaid primary conditions are satisfied upon the algorithm's iteration. The exact statistical sequence of

TABLE 4.12: Patterns D9–D10

Type of selected statistics	EOL_n (selection of the results' element) WL_n (website link access) mhC (mouse hovering activity over COE field)
Main conditions	D9-pattern: $\sum_{i=1}^{10} EOL_i = 0mhC$ D10-pattern: $\sum_{i=1}^{10} EOL_i = 0WL$
Secondary conditions	All actions are allowed between the specified mouse actions.

the given D8-pattern is represented as:

$$\begin{gathered} 3(EOL_n + AR_n) \\ 3(EOL_n + SR_n) \end{gathered} \tag{4.14}$$

The D8-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The second set of conditions specifies that all types of intermediate actions, performed by a user between the EOL selection and AR mouse click or SR slider movement are initially allowed, excluding only a website link access (WL). It should be noted that the D8-pattern will be broken or interrupted, if the user's off-pattern action of the website access is mixed with the pattern's initial set of statistical combinations.

Patterns D9–D10. The given two behavioural patterns, entitled as *D9* and *D10* are very similar in their compound structure and statistical conditions. Both patterns contain one element in common, that is a mouse action of the EOL selection. Additionally, each pattern contains its own specific statistical element, for the D9-pattern it is a mouse hovering activity over the COE field (mhC) and for the D10-pattern it is a website link access (WL). Collectively, all patterns' elements are regarded as a context-based statistical data (see Table 4.12).

Conventionally, both patterns consist of a series of mouse clicks, executed upon the elements of the results' list (D9, D10). Additionally the pattern D10 contains an instance of a mouse click upon the corresponding link of the selected result's website. On the other hand, the pattern D9 contains its own specific type of a mouse activity, defined as the mouse hovering over the COE field. The initial objective of the D9-pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting a pattern of a user's zero mouse hovering activity, within the limit of ten EOL selections.

The characteristics of the D10-pattern are somewhat similar to the D9 in a way that the D10-pattern's objective is to detect the case of a zero time of website accesses, within the same limit of ten EOL selections. The both patterns are marked completed, if the aforesaid primary conditions are respectively satisfied upon the algorithm's iterations. The exact statistical sequence of the D9-pattern is represented as:

$$\sum_{i=1}^{10} EOL_i = 0mhC \tag{4.15}$$

The exact statistical sequence of the D10-pattern is respectively represented as:

$$\sum_{i=1}^{10} EOL_i = 0WL \tag{4.16}$$

Alternatively, we may delineate the user's browsing activity, representing the results' selections as an interval between the first and tenth EOL selections. The respective alternative formulations for both patterns are:

D9-pattern: $$[EOL_1,\ EOL_{10}] = \{x \mid EOL_1 \leq x \leq EOL_{10},\ x = 0mhC\} \tag{4.17}$$

D10-pattern: $$[EOL_1,\ EOL_{10}] = \{x \mid EOL_1 \leq x \leq EOL_{10},\ x = 0WL\} \tag{4.18}$$

The patterns' main conditions are represented by a variable x, which delineates the required number of the detected mhC and WL_n context actions between the user's results selections of EOL_1 and EOL_{10}.

The D9 and D10-patterns' flexibilities in relation to their final completion are stipulated by the second set of algorithm's conditions. The second set of conditions for both patterns specifies that all types of intermediate actions, performed by a user between the specified mouse actions are initially allowed. It should be noted that the logic of both patterns will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

All composite mini patterns from D1 to D10, collectively represent a complete cognitive model of the user's emotional state of dissatisfaction. Every element of the model plays an important role in an overall process of the model detection due to its unique capacity of representing on an algorithmic level the particular singular instances of a user's browsing activity.

The given model is defined as a non-static sequence of the accordingly allocated statistical data, which has the objective to represent and to reflect the user's emotional state of *discontent*. From a conceptual point of view, the model is detected progressively, by means of a sequential or random detection of the model's composite mini patterns, represented in form of specific statistical combinations (D1–D10). The exact mechanism of the model detection, will be successively delineated in the upcoming section (see section 22). Before proceeding to the description of the model detection mechanism, we present first in the next section the corresponding cognitive model of a user's satisfaction.

4.2.5 Satisfaction model

Our second cognitive model, entitled as *Satisfaction model* (S-model), aims to represent on an algorithmic level the user's opposite cognitive state of satisfaction, manifested in relation to the currently displayed search results. The simulation of the given state is based on a detection of user's overtly manifested instances of an emotional contentment, which are stipulated by the context conditions of relevant or high relevant results.

The cognitive state of satisfaction is identified by means of detecting the instances of a user's browsing consistency, manifested throughout the sessions of a search querying. The satisfaction model is based on the "indicators" of high relevant results, which have the initial objective to signal the potential presence of a user's mental state of an emotional contentment.

For the purposes of achieving the aforementioned goal, we have to quantitatively define the abstract problem of a mental satisfaction in terms of user's manifested context actions. Consequently, in order to represent the particular instances of a user's overt behaviour on the algorithmic level, we must similarly allocate all collected statistical data into the specific computational patterns.

For the purposes of designing the model's compound mini patterns, we elaborate a set of supporting guidelines, allowing subsequently to facilitate the process of the patterns definition. The satisfaction model consists of several key principles, which define the main directions of the user's pattern specifications. The given directions are essentially characterised by the user's more extensive interactions with the results, considerably larger time of the results analysis, minimal number of skipped results' elements and the user's more consistent mouse hovering activity over the COE field. The modelling process of user's behavioural patterns is fully based on the given guidelines of most probable types of context actions.

Table 4.13: Pattern S1

Type of selected statistics	EOL_n (selection of the results' element) AR_n (sidebar arrow click) SR_n (sidebar slider movement)
Main conditions	$\left.\begin{array}{c}[AR_1, AR_2]\\ [AR_1, SR_2]\\ [SR_1, AR_2]\\ [SR_1, SR_2]\end{array}\right\} = \{x \mid x \geq 4EOL\}$
Secondary conditions	All actions are allowed between the specified mouse actions.

Similarly to the D-model, all mini patterns of the S-model represent a fully functional cognitive model, which is stored in a segregated form inside of the same context monitoring scenario MTA-c. In total, the satisfaction model contains equally ten mini patterns of the correlated statistical data (S1–S10), represented in form of sequential algorithmic steps. In the upcoming paragraphs, we present the model's composite patterns, by highlighting their internal algorithmic structures, corresponding abstract meanings and the character of their functions in relation to the full model. The position of the composite patterns, in relation to their number (S1–S10) does not have any special structure or logic, all patterns are enumerated with regard to a process of their gradual design.

Pattern S1. The given behavioural pattern, entitled as *S1* initially contains three statistical elements, that is a mouse action of the EOL selection, AR click and SR movement. Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.13). The S1-pattern consists of a series of mouse clicks, executed respectively upon the elements of the results' list (EOL_n) and the results' list scrollbar either by means of a sidebar arrow click (AR_n) or a sidebar slider movement (SR_n).

The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting minimum four EOL selections between any combinations of the scrollbar activity (AR_n/SR_n). The algorithm is first set to detect the opening action of either the sidebar arrow click (AR_1), or the sidebar slider movement (SR_1).

Subsequently, the algorithm has the objective to detect minimum four EOL selections ($4EOL$), executed sequentially or randomly, with a possibility of user's actions deviation from the predefined pattern's structure. The pattern is completed by the following closing action of either the sidebar arrow click (AR_2), or the sidebar slider movement

Table 4.14: Pattern S2

Type of selected statistics	EOL_n (selection of the results' element) WL_n (website link access)
Main conditions	$EOL_n + WL_n$
Secondary conditions	All actions are allowed between EOL_n selection and website link access (WL_n), except mouse hovering activity over COE field (mhC). If mhC is detected then count as S3-pattern.

(SR_2). The pattern is marked fully completed, if the aforesaid primary conditions are satisfied upon the algorithm's iteration.

For the purposes of a concise delineation, we present the four possible variations of the scrollbar activity as the combinations of four different intervals between the opening and closing actions [AR_1/SR_1, $AR_2/SR2$]. The exact statistical sequence of the S1-pattern is therefore represented as a combination of various intervals, stipulated by the common context condition of x:

$$\left.\begin{array}{c}[AR_1, AR_2]\\ [AR_1, SR_2]\\ [SR_1, AR_2]\\ [SR_1, SR_2]\end{array}\right\} = \{x \,|\, x \geq 4EOL\} \tag{4.19}$$

As we may see, the pattern's main condition is represented by a variable x, which delineates the minimum required number of the selected results' elements between the mouse actions of AR_1/SR_1 and AR_2/SR_2.

The S1-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions specify that all types of intermediate actions, performed by a user between the specified mouse actions are initially allowed. The S1-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

From an abstract point of view, the S1-pattern aims to represent the higher level of the browsing consistency in user's search querying attempts. During a search activity, conditioned by the presence of high relevant results, the user tends to examine more thoroughly every results' entry in comparison to the skimming type of browsing in the case of low relevant results.

TABLE 4.15: Pattern S3

Type of selected statistics	EOL_n (selection of the results' element) WL_n (website link access) mhC (mouse hovering activity over COE field)
Main conditions	$EOL_n + mhC + WL_n$
Secondary conditions	All actions are allowed between EOL_n selection and mhC activity, except WL_n click. All actions are allowed between mhC activity and WL_n click, except EOL_n selection.

Pattern S2. The given behavioural pattern, entitled as *S2* initially contains only two statistical elements, that is a mouse action of the EOL_n selection and website link access (WL_n). Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.14). The S2-pattern consists of a series of mouse clicks, executed respectively upon the elements of the results' list (EOL_n) and a corresponding website link (WL_n). The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting a simple combination of a browsing action, defined as $EOL_n + WL_n$. The algorithm is first set to detect the opening action of EOL_n selection, followed by a closing context action of the website link access (WL_n). The pattern is marked fully completed, if the aforesaid primary conditions are satisfied upon the algorithm's iteration. The S2-pattern's composite statistical combination is represented by a simple data sequence:

$$EOL_n + WL_n \tag{4.20}$$

The S2-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions of the given pattern are somewhat more complex in relation to the conditions of all previous patterns. The second set of conditions specifies that all types of intermediate actions, performed by a user between the EOL_n selection and WL_n access are initially allowed, except the mouse hovering activity over the COE field (mhC). If the mhC data has been detected during the algorithm's iteration, the given statistical sequence is then marked and accordingly accounted as the S3-pattern.

Pattern S3. The given behavioural pattern, entitled as *S3* is almost identical in its structure to the preceding pattern S2 with a small difference of the additional statistical element (see Table 4.15). The S3-pattern contains three statistical elements, that is a mouse action of the EOL_n selection, website link access (WL_n) and a mouse hovering activity over the COE field (mhC). With regard to the additional statistical element of the mhC, the given S3-pattern represents an extension of the S2-pattern. The key factor for differentiating the two patterns is stipulated by our intention to reveal and to account the user's mouse hover activity over the COE field. Upon its iteration, the algorithm is first set to detect the opening action of the EOL_n selection, followed by an intermediate action of the mouse hovering (mhC) and ultimately by a closing context action of the website access (WL_n). The pattern is marked fully completed, if the aforesaid primary conditions are satisfied upon the algorithm's iteration. The S3-pattern's composite statistical combination is represented by a simple data sequence:

$$EOL_n + mhC + WL_n \tag{4.21}$$

The S3-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. In order to be able to reveal the user's mouse hovering activity, the secondary conditions are specific to the pattern's every composite segment. On the one hand, that are the condition requirements for the first segment of $[EOL_n + mhC]$ and, on the other hand, that are the specific condition requirements for the second pattern's segment of $[mhC+WL_n]$. In case of the first segment, the conditions stipulate that all actions are allowed between the results' selection and mouse hovering activity over the COE field (mhC), except the user's action of the website link access (WL_n). The conditions of the second pattern's segment stipulate that all actions are allowed between the mouse hovering activity over the COE field (mhC) and a user's action of the website access (WL_n), except the selection of the results' element (EOL_n). The given two sets of the secondary conditions allow us to reveal the exact sequence of the user's context actions. If one of the two sub-conditions is not satisfied upon the algorithm's iteration, the S3-pattern becomes broken, except the case of the first segment conditions, where the broken statistical combination automatically becomes the S2-pattern.

Pattern S4. The given behavioural pattern, entitled as *S4* initially contains three statistical elements, that is a mouse action of the EOL selection, time difference or total time of stay of the selected results' element ($\Delta t_{(EOL_w)_n}$) and the time difference, indicating an average time of stay of the selected results' element ($\Delta \bar{t}_{(EOL_w)_n}$). Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.16). The S4-pattern consists of a series of mouse clicks, executed upon the elements of the results' list (EOL_n). The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting the two set of conditions. Both sets aim to respectively detect (a) minimum four consecutive selections of the same website with an average time of stay minimum four seconds per

Table 4.16: Pattern S4

Type of selected statistics	EOL_w (selections of results' elements with the same website) $\Delta t_{(EOL_w)_n}$ (total time of stay of a selected EOL_w) $\Delta \bar{t}_{(EOL_w)_n}$ (average time of stay of a selected EOL_w)
Main conditions	$\Delta t_{(EOL_w)_n} = t_{(EOL_w)_{n+1}} - t_{(EOL_w)_n}$ (a) $\Delta \bar{t}_{(4EOL_w)} = \frac{1}{4}\sum_{n=1}^{4} \Delta t_{(EOL_w)_n} \geq 4sec$ (b) $\sum EOL_i \geq 8EOL_w$
Secondary conditions	All actions are allowed between every EOL_w selection.

EOL ($\Delta \bar{t}_{(4EOL_w)} \geq 4sec$) and (b) if there are minimum eight nonconsecutive selections of the same website in the overall user's activity ($\sum EOL_i \geq 8EOL_w$).

In case of the first series of conditions (a), the algorithm is set to detect four consecutive results' selections from the same online source ($4EOL_w$). Subsequently, the algorithm has the objective to calculate the average time of stay for the given four elements, which conventionally has to be minimum 4 seconds per EOL ($\Delta \bar{t} \geq 4sec$). In order to calculate the given time conditions, we must first specify, by means of a simple time deduction the time difference, or the total time of stay for every selected results' element ($\Delta t_{(EOL_w)_n}$). After having specified the total time of stay for every four selected EOL, we calculate furthermore the values of their average time of stay ($\Delta \bar{t}_{(4EOL_w)}$).

Similarly to the D2-pattern, for the purposes of calculating the average, we use a simple arithmetic mean equation (4.3). We adapt the aforementioned equation in order to calculate the average time of stay for every four selected results' elements. Additionally, we specify the related context-based conditions, which stipulate that the calculated average time of duration for every selected EOL has to be equal minimum to 4 seconds:

$$\Delta \bar{t}_{(4EOL_w)} = \frac{1}{4}\sum_{n=1}^{4} \Delta t_{(EOL_w)_n} \geq 4sec \tag{4.22}$$

In case of the second set of the pattern's conditions (b), the algorithm is simply programmed to detect every EOL selection, performed by a user in order to reveal the potential eight results' elements of the same type. The algorithm calculates, if in the overall user's browsing activity there are at least eight nonconsecutive EOL selections from the same online source ($8EOL_w$). We delineate the user's overall browsing activity

TABLE 4.17: Pattern S5

Type of selected statistics	EOL_w (selections of results' elements with the same website) mhC (mouse hovering activity over COE field)
Main conditions	$3(EOL_w + mhC)$
Secondary conditions	All actions are allowed between EOL_w selection and mhC activity. All actions are allowed between the sequences of a data detection.

within the results' list as an unconditioned sum of the randomly selected results' elements ($\sum EOL_i$). The second part of the S4-pattern is marked fully completed, if the aforementioned set of general conditions is satisfied upon the algorithm's iteration. The exact sequence of the given statistical combination is represented as:

$$\sum EOL_i \geq 8EOL_w \tag{4.23}$$

The S4-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions are individual to both sets of primary conditions. Precisely, for the first data sequence (a), the secondary conditions specify that all actions are allowed between the selections of four consecutive EOL, excluding the selection of an EOL with the different online source. The first sequence of the S4-pattern will not be broken, unless the user explicitly selects an EOL, corresponding to a different online source in comparison to the previously selected results. For the second data sequence (b), the secondary conditions specify that all actions are allowed between the nonconsecutive selections of EOL, including the selection of an EOL with the different online source.

Pattern S5. The given behavioural pattern, entitled as *S5* initially contains two statistical elements, that is a mouse action of the EOL selection and a mouse hovering activity over the COE field. Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.17). The S5-pattern consists of a series of mouse clicks, executed respectively upon the elements of the results' list (EOL_n) and a mouse hovering activity, performed by a user over the COE field.

The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting consecutively or nonconsecutively minimum three times the same browsing pattern. The pattern consists of

Table 4.18: Pattern S6

Type of selected statistics	EOL_t (selections of results' elements with the same tag name) mhC (mouse hovering activity over COE field)
Main conditions	$2(EOL_t + mhC)$
Secondary conditions	All actions are allowed between EOL_t selection and mhC activity. All actions are allowed between two sequences of a data detection.

one EOL selection, associated with the same online source and a mouse hovering action, performed by a user over the COE field. The algorithm is first set to detect the opening action of an EOL selection of a particular website (EOL_w), followed by a closing action of the mouse hovering activity over the COE field (mhC). Subsequently, the algorithm has the objective to monitor the user's actions for two more similar sequences of the same data types. The S5-pattern is completed only if all three data sequences, detected consecutively or non-consecutively, contain an EOL element, associated with the same online source. The pattern is marked fully completed, if the aforesaid primary conditions are satisfied upon the algorithm's iteration. The exact structure of the S5-pattern is represented as a simple twofold statistical sequence of the allocated context data:

$$3(EOL_w + mhC) \tag{4.24}$$

The S5-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions specify that all types of intermediate actions, performed by a user between the EOL selection and a mouse hovering activity are initially allowed. Furthermore, the conditions specify that all types of context actions are allowed between three sequences of the allocated context data. The S5-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern S6. The given behavioural pattern, entitled as *S6* structurally is similar to the previous S5-pattern. The only major difference is represented by the algorithm's objective of detecting the EOL selections, associated not with the same website, but with the same tag name. Therefore, the given pattern contains the same statistical elements, that is a mouse action of the EOL selection and a mouse hovering activity over the COE field (see Table 4.18). The S6-pattern consists of a series of mouse clicks,

TABLE 4.19: Pattern S7

Type of selected statistics	EOL_t (selections of results' elements with the same tag name)
Main conditions	$\sum EOL_i \geq 3EOL_t$
Secondary conditions	All actions are allowed between EOL_t selections.

executed respectively upon the elements of the results' list (EOL_n) and a mouse hovering activity over the COE field.

The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting, consecutively or non-consecutively, minimum two times the same browsing pattern. The pattern consists of one EOL selection, associated with the same tag name and a mouse hovering action, performed by a user over the COE field. The algorithm is first set to detect an opening action of the EOL selection of a particular tag (EOL_t), followed by a closing action of the mouse hovering activity over the COE field (mhC). Subsequently, the algorithm has the objective to monitor the user's actions for one more similar sequence of the same data types.

The S6-pattern is completed only if both data sequences, detected consecutively or non-consecutively, contain an EOL element, associated with the same tag name. The pattern is marked fully completed, if the aforesaid primary conditions are satisfied upon the algorithm's iteration. The exact structure of the S6-pattern is represented as a simple twofold statistical sequence of the allocated context data:

$$2(EOL_t + mhC) \tag{4.25}$$

The S6-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions specify that all types of intermediate actions, performed by a user between the EOL selection and a mouse hovering activity are initially allowed. Furthermore, the conditions specify that all types of context actions are allowed between two sequences of the allocated context data. The S6-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern S7. The given behavioural pattern, entitled as *S7* initially contains only one statistical element, that is a mouse action of the EOL_t selection (see Table 4.19). The S7-pattern consists of a series of mouse clicks, executed upon the elements of the

TABLE 4.20: Pattern S8

Type of selected statistics	EOL_n (selection of the results' elements) mhC (mouse hovering activity over COE field)
Main conditions	$\sum_{i=1}^{10} EOL_i \geq 3mhC$
Secondary conditions	All actions are allowed between EOL_n selections.

results' list. The initial objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting minimum three EOL_t selections with the same tag name in an overall user's activity. The algorithm is set to detect the first EOL_t element, followed by the detection of two more EOL_t elements with exactly the same tag name. The pattern is marked fully completed, if the aforesaid primary conditions are satisfied upon the algorithm's iteration. We represent the user's overall browsing activity within the results' list as an unconditioned sum of randomly selected results' elements ($\sum EOL_i$). The exact sequence of the given statistical combination is represented as:

$$\sum EOL_i \geq 3EOL_t \tag{4.26}$$

Alternatively, we may delineate the user's browsing activity, representing the process of overall results' selections as an interval between 1^{st} and n^{th} EOL selections:

$$[EOL_1,\ EOL_n] = \{x \,|\, EOL_1 \leq x \leq EOL_n,\ x \geq 3EOL_t\} \tag{4.27}$$

The pattern's main condition is represented by a variable x, which delineates the minimum required number of the detected EOL elements with the same tag names between user's result selections of EOL_1 and EOL_n.

The S7-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions specify that all types of intermediate actions, performed by a user between the EOL selections are initially allowed. The S7-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern S8. The given behavioural pattern, entitled as *S8* initially contains two statistical elements, that is a mouse action of the EOL_n selection and a mouse hovering activity over the COE field. Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.20). The S8-pattern consists of a series of mouse clicks,

TABLE 4.21: Pattern S9

Type of selected statistics	EOL_n (selection of the results' elements) WL_n (website link access)
Main conditions	$\sum_{i=1}^{10} EOL_i \geq 3WL$
Secondary conditions	All actions are allowed between EOL_n selections.

executed upon the elements of the results' list. The objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting minimum three mhC activities within the total number of ten selected EOL_n. The algorithm is set to detect at least three mouse hovering actions, performed by a user within an overall process of the results' browsing, which is conditioned by the selection of ten EOL elements. We represent the user's browsing activity within the results' list as a conditioned sum of randomly selected results' elements ($\sum EOL_i$). The exact sequence of the given statistical combination is represented as:

$$\sum_{i=1}^{10} EOL_i \geq 3mhC \tag{4.28}$$

Alternatively, we may delineate the user's browsing activity, representing the results' selections as an interval between first and tenth EOL selections:

$$[EOL_1,\ EOL_{10}] = \{x \,|\, EOL_1 \leq x \leq EOL_{10},\ x \geq 3mhC\} \tag{4.29}$$

As we may notice, the pattern's main condition is represented by a variable x, which delineates the minimum required number of mouse hovering actions between the user's result selections of EOL_1 and EOL_{10}.

The S8-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions specify that all types of intermediate actions, performed by a user between the EOL selections are initially allowed. The S8-pattern will not be broken or interrupted if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern S9. The current behavioural pattern, entitled as *S9* is similar to the preceding S8-pattern with the difference that the S9-pattern contains a statistical element of the website access instead of a mouse hovering activity. Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.21). The S9-pattern

Table 4.22: Pattern S10

Type of selected statistics	EOL_n (selection of the results' element) T_n (a uniform tag-set $\{a, a\}$)
Main conditions	$\sum_{i=1}^{15} EOL_i = 1T_n$
Secondary conditions	All actions are allowed between EOL_n selections.

consists of a series of mouse clicks, executed respectively upon the elements of the results' list. The objective of the given pattern is to monitor the user's interactions with various elements of the results' page for the purposes of detecting minimum three website accesses within the total number of ten selected EOL_n. The algorithm is set to detect at least three accesses to the corresponding online sources, performed by a user within an overall process of the results' browsing, which is conditioned by the selection of ten EOL elements. We represent the user's browsing activity within the results' list as a conditioned sum of randomly selected results' elements ($\sum EOL_i$). The exact sequence of the given statistical combination is represented as:

$$\sum_{i=1}^{10} EOL_i \geq 3WL \tag{4.30}$$

Similarly to the preceding S8-pattern, we may delineate the user's browsing activity, representing the results' selections as an interval between the first and tenth EOL selections:

$$[EOL_1,\ EOL_{10}] = \{x \mid EOL_1 \leq x \leq EOL_{10},\ x \geq 3WL\} \tag{4.31}$$

The pattern's main condition is represented by a variable x, which delineates the minimum required number of website accesses between the user's result selections of EOL_1 and EOL_{10}.

The S9-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions specify that all types of intermediate actions, performed by a user between the EOL_n selections are initially allowed. The S9-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

Pattern S10. The final behavioural pattern, entitled as *S10* is similar to two preceding pattens of the S-model and to the D7-pattern of the D-model. The current S10-pattern contains two statistical elements, that is a mouse action of the EOL_n selection and a content related metadata, which describes a type of the selected EOL_n

by a corresponding content tag. As we defined earlier in (4.9), the given metadata is represented by a uniform tag-set (T_n). Collectively, all pattern's elements are regarded as a context-based statistical data (see Table 4.22). The pattern consists of a series of mouse clicks, executed upon the elements of the results' list.

The objective of the given pattern is to monitor the user's interactions with various elements of the results' list for the purposes of detecting the pattern of one uniform tag-set ($1T_n$) within the limit of fifteen EOL selections ($\sum_{i=1}^{15} EOL_i$). In order to detect the given data, the algorithm has to continuously take a record of every selected EOL's metadata. The pattern is marked completed, if a sum of fifteen selected results' elements contains at least one uniform tag-set. For the purposes of detecting the given pattern, the algorithm has the objective to monitor the sets of every fifteen EOL selections for the presence of a distinctive metadata. If minimum one uniform tag-set has been detected, the pattern is marked as completed. The exact sequence of the S10-pattern is represented as:

$$\sum_{i=1}^{15} EOL_i = 1T_n \tag{4.32}$$

Similarly to the preceding two patterns, we may delineate the user's browsing activity, representing the results' selections as an interval between first and fifteenth EOL selections:

$$[EOL_1,\ EOL_{15}] = \{x \,|\, EOL_1 \leq x \leq EOL_{15},\ x \geq 1T_n\} \tag{4.33}$$

The pattern's main condition is represented by a variable x, which delineates the minimum required number of the detected uniform tag-sets between the user's result selections of EOL_1 and EOL_{15}.

The S10-pattern's flexibility in relation to its final completion is stipulated by the second set of algorithm's conditions. The secondary conditions specify that all types of intermediate actions, performed by a user between the EOL_n selections are initially allowed. The S10-pattern will not be broken or interrupted, if one of the user's off-pattern actions is mixed with the pattern's initial set of statistical combinations.

All presented composite mini patterns from S1 to S10, collectively represent a complete cognitive model of the user's emotional state of satisfaction. Every element of the model plays an important role in an overall process of the model detection due to its unique capacity of representing on an algorithmic level the particular singular instances of a user's browsing activity. The given model is defined as a non-static sequence of the accordingly allocated statistical data, which has the objective to represent and to reflect the user's emotional state of *contentment*. From a conceptual point of view, similarly to the D-model, the S-model is detected progressively by means of a sequential or random detection of the model's composite mini patterns, represented in form of the specific statistical combinations (S1–S10). The exact mechanism of the model detection, will be successively delineated in the upcoming section (see section 22).

4.3 Threefold structure of the system's probabilistic framework

In order to allow the system to accurately detect the composite details of a context, to perform the analysis of the detected data and ultimately to provide the required proactive mediation, we implement a threefold mechanism of the system's probabilistic framework. As we will see in the present section, the proactive system's mechanism is based on the operation of three proactive scenarios, MTA-c, MTA-h and MTA-m. Collectively, all proactive modules provide the functional basis of the proactive system on the corresponding probabilistic level.

4.3.1 Progressive mechanism of a model detection (MTA-c)

Our first Meta-scenario, denoted as *MTA-c* represents the main proactive system's instrument, allowing to translate the abstract domain of the user's prevailing cognitive characteristics into an algorithmic level of computational processes (see Algorithm 22). The given proactive scenario represents the search engine's context monitoring mechanism, which keeps the manifestations of a user's context behaviour under continuous observation. As soon as the predefined instance of the user's browsing behaviour is manifested, the MTA-c scenario is set to re-evaluate the overall context situation with respect to a type of the newly detected data instance. In other words, the given MTA-c scenario contains in its algorithmic structure our all previously discussed mini patterns (statistical combinations) of the D-model and S-model. The MTA-c scenario is therefore our main context-aware mechanism of detecting the corresponding cognitive states of a user during his/her search activity.

The mechanism of a pattern detection consists of a continuous monitoring of the proactive system's database, where all context-based statistics of the user's browsing actions are stored by the system. As soon as a new context data is detected, the algorithm of the MTA-c scenario is set to check if the given data corresponds to the characteristics of one of its composite model's patterns. Upon condition that one of the patterns is completed, the algorithm successively re-evaluates the conditions of the currently present context settings in order to reflect the accurateness of its actual state. As we will see in details below, the mechanism of the patterns' completion for both models, as well as the probabilistic evaluation of current context conditions are entirely based on the deterministic principles of the scenario's rule structure.

The MTA-c scenario as well as all subsequently presented proactive scenarios are built upon, and fully based on the previously presented rule's template, which represents the general model-structure of the proactive scenario's algorithm. From a programming point of view, such approach implies that the inclosed scenario's algorithm inherently consists of five rule's elements. Each element has its specific functions and corresponding parameters.

Algorithm 22: MTA-c scenario

```
Input: N - last event treated by this rule in a previous iteration.
Data Acquisition()
    boolean newEvents ← isNewEvent(N);
Activation Guards()
    return newEvents;
Conditions()
    return true;
Actions()
    nextN ← last event ID;
    Read events from DB (in reverse order);
    for (each event) do
        Get event ID;
        Detect event type;
        for (each pattern) do
            Check each pattern against event type; // with respect to nextN and
            detected event
            if (pattern is found) then
                Recalculate model prevalence;
                Assign nextN to the pattern;
                Save the pattern in DB;
                Re-initialise the pattern;
Rule Generation()
    if (getActivated) then
        createRule MTA-c(nextN)
    else createRule MTA-c(N);
```

At the level of the first algorithm's step of *Data acquisition*, the rule's function is set to comparatively evaluate the database state for the purposes of detecting a new event in form of a distinct statistical instance. All manifestations of a user's overt behaviour, expressed within the search engine's environment are represented in form of the predefined statistical combinations, or in other words behavioural patterns. Therefore, in order to detect a specific behavioural pattern, the algorithm is set first to detect all pattern's constituent elements, represented by the individual instances of a statistical data. For the given purposes, the corresponding function of the Data acquisition part is set to detect every new instance of a context data ($newEvent$), stored in the proactive system's database. In order to identify the new data, the function uses as a parameter the last event, treated by the given rule in the previous iteration (parameter N). After having detected the new event on the database, the algorithm's control flow is subsequently directed through the *Activation guards* and *Conditions* towards the rule's fourth step of *Actions*.

The given fourth part of the algorithm's logic has the objective to evaluate the new data with respect to the compound characteristics of each behavioural pattern. The functions of the algorithm's actions-part are primarily set to collect all necessary characteristics of the detected event, required by the successive functions of the given scenario. The collected event characteristics, extracted from the corresponding database tables include such information as an event ID and event type. Consequently, all twenty composite behavioural patterns, which are stored in the structure of the Meta-scenario as array objects are evaluated against a newly detected event type. The evaluation of a pattern's structure against the detected event is performed with respect to the parameter of the last event's ID ($nextN$), which allows the algorithm to separate the pattern, once it is completed. The application of the $nextN$ parameter plays an important role in a pattern segregation, which consequently allows the algorithm not to repeat the completed pattern upon an evaluation of the next detected event.

It should be noted that a new event evaluation takes place individually within the algorithmic structure of each pattern. After the new event has been detected by earlier functions of the MTA-c scenario, the detected event's characteristics are subsequently passed to each pattern as a parameter for the purposes of self-evaluation. Correspondingly, every pattern is set to assess its current state with regard to the newly detected context event. The main objective of each pattern is therefore to identify if a type of a newly received statistical data corresponds to the pattern's internal composition and if it allows the pattern to update its current state. In other words, every pattern evaluates individually, within itself, the characteristics of a newly detected event.

Upon the detection of a new event and depending on the logic of the given pattern, the current state evaluation includes several corresponding possibilities. (1) If the newly received context data has zero relevance to the pattern, the current state of the given pattern remains unchanged. (2) If the newly received context data breaks the logic of the pattern, the current state of the given pattern resets to its initial value. (3) If the newly received context data is relevant to the pattern, the current state of the given pattern gets consequently updated by means of advancing the pattern's status one step closer to its final state of the completion. Upon condition that the pattern, throughout the process of its update, reaches the final state of the completion, it subsequently returns the *true* value to the MTA-c scenario for the purposes of a successive activation of the model's evaluation function.

In order to evaluate the prevalence level for each model, we apply the Bayes' theorem of the conditional probabilities, which allows us to perform the statistical inference between two competing states of the user's current mental characteristics. The Bayes' theorem is mathematically represented as follows:

$$P(H|S) = \frac{P(S|H)\,P(H)}{P(S)} \tag{4.34}$$

The denotation of $P(H|S)$ may simply be translated as a probability of the hypothesis H, given that the event S has already occurred. In order to adapt the Bayes' rule to our case, we may rephrase the given statement as a probability of the hypothesis that the ***H****igh relevant results* are currently present (H), given that the ***S****atisfaction model* prevails (S). According to the Bayes' rule, the conditional probability of $P(H|S)$ can be acquired by dividing the product of the *Likelihood function* $P(S|H)$ and *Prior probability* $P(H)$ by a sum of the *Joint probabilities* $P(S)$. In the similar manner, we may formulate our second hypothesis, which deals with the odds of the *Dissatisfaction model*. The given formulation has the following form:

$$P(L|D) = \frac{P(D|L)\,P(L)}{P(D)} \tag{4.35}$$

The denotation of $P(L|D)$ may equivalently be translated as a probability of the hypothesis that the ***L****ow relevant results* are currently present (L), given the fact that the ***D****issatisfaction model* prevails (D). Consequently, both formulas represent the interdepend relationship between the ***H****igh relevant results* (H) and ***L****ow relevant results* (L). The initial values, which represent the ratio between H and L are set respectively to 50% for each type of the prior probabilities. Subsequently, we specify the parameters for the rest of our probabilistic model, which allow the algorithm to perform an estimate of a cognitive model's prevalence upon every update of the initial prior probability distribution.

As indicated in Table 4.23, the values between different parts of the model are represented as the simple proportional relations, whose sum is always equal to 100%, including the ratio of 50% and 50% between H and L, and the ratio of 70% and 30% between S

Table 4.23: Probabilistic model values

$P(H)$	50%
$P(L)$	50%
$P(S\|H)$	70%
$P(D\|H)$	30%
$P(S\|L)$	30%
$P(D\|L)$	70%

and D. In case of the H-branch of the probabilistic model, we have 70% for $P(S|H)$ and 30% for $P(D|H)$. In case of the L-branch, we have respectively 30% for $P(S|L)$ and 70% for $P(D|L)$.

The Bayes' formula functions as the updater mechanism for the initial values of the prior probability distribution (H and L). Each time a completed pattern of one of the cognitive models returns the *true* value to the MTA-c scenario, the Bayes' rule updates the initial ratio of 50% : 50% between the ***H**igh relevant results* hypothesis and ***L**ow relevant results* hypothesis. In other words, depending on a type of the completed pattern, which may be a part either of the ***S**atisfaction* or ***D**issatisfaction* model the Bayes' rule consequently recalculates the initial values of the prior probability distribution. From a theoretical point of view, if $H > L$, then the S-model and user's cognitive state of *satisfaction* currently prevail in relation to the progress level of the D-model. Correspondingly, if $L > H$, then the D-model and user's cognitive state of *dissatisfaction* currently prevail in relation to the progress level of the S-model. After every update of the prior probability distribution, the corresponding resulted values of the proportional relations, which indicate the prevalence level of each cognitive model are stored in the proactive system's database.

Consequently, after the recalculation of the models' prevalence, the algorithm of the MTA-c scenario assigns the *nextN* value to the detected pattern and saves it in the database. Upon the last step of the actions-part, the algorithm is set to re-initialise the completed pattern to its initial state. The given approach allows the algorithm to detect the current behavioural pattern again, if it is successively manifested by a user during his/her future search querying attempts.

According to the characteristics of the previously discussed proactive rule's structure, upon its last fifth step the algorithm of the MTA-c scenario is set to generate a new instance of itself in order to sustain the functional aspect of the process continuity. The generated instance of the rule is parametrised accordingly, depending on a resulted value of the preceding Activation guards.

From a conceptual point of view, the given MTA-c scenario represents the proactive system's main mechanism, allowing to perform the continuous monitoring of the search engine's context environment. By means of the given proactive scenario, the system is capable, through the detection of the predefined behavioural pattern, to observe the characteristics of a user's browsing activity. In comparison to the next two proactive scenarios, the presented below MTA-c scenario may be considered as the most complex in terms of its structure and constituent composite elements.

The complexity of the MTA-c scenario is mainly characterised by the presence of two computational factors. The first factor represents the design choice of placing all twenty patterns of the D-model and S-model inside the structure of the MTA-c scenario. The second factor is stipulated by the design choice of implementing the module of the statistical analysis, conceptually represented by the function of the Bayesian parameter estimation. Consequently, the MTA-c scenario represents a unique proactive mechanism, allowing on the one hand, to detect the predefined instances of a context data and, on the other hand, to analyse the collected data in order to identify its finer metaphysical properties in form of the underlying abstract meanings. The given Meta-scenario, represents our main technique for identifying the hidden context aspects, which are mainly characterised by the user's prevailing cognitive states.

4.3.2 Proactive mechanism of a history monitoring (MTA-h)

In order to implement the system's proactive behaviour and thus to apply the corresponding context actions, related to a user's browsing activity, the Meta-scenario MTA-h is set to analyse the history of user's context actions, through a continuous monitoring of corresponding database tables. The algorithm of the given scenario has the objective to monitor the progression state of two earlier presented pattens of the S-model, that is the S2 and S3 statistical combinations. The principles of the MTA-h scenario stipulate that as soon as one or both patterns are completed, the algorithm will successively initiate the analysis of the user's history actions. The provided information by the S2 and S3 patterns is used by the MTA-h scenario as the basis for identifying similar, non-viewed alternative results (see Algorithm 23). As we will see later, the identification of alternative results is based on the principles of the data association.

At the level of the first algorithm's step of *Data acquisition*, the rule's function is set to comparatively evaluate the database state and to detect, if the pattern S2 and/or S3 have been completed. Additionally, the algorithm has the objective to detect the very first EOL, selected by a user right after the composite EOL of the patterns S2 and S3. As indicated in the pseudocode, the detection of the new EOL, successively serves as a

Algorithm 23: MTA-h scenario

Input: N_1 - last pattern treated by this rule in a previous iteration.

Input: N_2 - last EOL treated by this rule in a previous iteration.

```
Data Acquisition()
    boolean newS2pattern ← isNewS2completed(N_1);
    boolean newS3pattern ← isNewS3completed(N_1);
    boolean newEOL ← isNewEOL(N_2);
Activation Guards()
    return newS2pattern, newS3pattern, newEOL;
Conditions()
    return true;
Actions()
    if (STATE==searching) then
        if (S2 or S3 have been detected) then
            create currentSet ← {e_1, e_2, null}; // e_1=tag(EOL_{n-1}), e_2=tag(EOL_n)
        foreach (i=Set_1;  i=currentSet-1; i++) do
            if ((i has e_1) or (i has e_2)) then
                if (i has max. 2e_1) then
                    remainTags1 ← i - e_1; // identifying associated tags
                if (i has max. 2e_2) then
                    remainTags2 ← i - e_2
                if (i has max. 1e_1 and 1e_2) then
                    remainTags3 ← i - (e_1 + e_2)
                collectedTags ← {remainTags1, remainTags2, remainTags3};
                collectedTags ← currentSet
        create obtainedResults ← SearchFor(collectedTags).Within(DB);
        alternativeResults ← {obtainedResults};
    if (STATE==closing) then
        complete currentSet ← {e_1, e_2, e_3};
        Set_n ← currentSet; // Set_n[] - last completed set
Rule Generation()
    if (getActivated and STATE==searching) then
        createRule MTA-h(nextN_2)
    if (getActivated and STATE==closing) then
        createRule MTA-h(nextN_1)
    if (notActivated) then
        createRule MTA-h(N_1)
```

condition for the purposes of closing the current set of the selected tags. Both types of a context data, specified in the Data acquisition part, subsequently direct the control flow to the corresponding part of the rule's actions. The given segment of the algorithm deals either with an identification of the associated tags, upon the detection of the S2 or S3 patterns, or it deals with an action of completing the current set of tags, upon the detection of a new selected EOL.

Consequently, the rule's fourth step of actions is divided in two algorithmic states. Each state is activated according to a type of the detected data by the rule's first step of the Data acquisition. Upon the detection of the completed S2 or S3 patterns, the first part, dealing with an identification of the associated tags is activated. The given part of the algorithm's actions has the objective first to identify and then to extract the associated results' tags from the search engine's database. Upon the activation of the actions part, the algorithm is set first to create the current set of the associated tags. The current set, conventionally consisting of the three elements, at the beginning of the process includes only two composite elements.

The main element, denoted as e_2 is represented by the "triggering" EOL, which makes part of the detected S2 or S3 pattern. The given triggering EOL (EOL_n) and its corresponding tag constitute the initial and very first element of the current set. The second element of the set, denoted as e_1 is an EOL, which has been selected one step before the initial EOL_n. In other words, the second element of the current set is the EOL_{n-1} and its corresponding tag. The third element of the current set is simply set to *null* value as at the given moment it represents the future, not yet happened selection of the third EOL (EOL_{n+1}). The third element of the current set is an EOL, which will be selected by a user right after the EOL_n. In other words, the current set, which has to be initialised by the algorithm at the moment of the detection of the S2 or S3 patterns may be represented as: $\{EOL_{n-1}, EOL_n, EOL_{n+1}\}$, where EOL_n is the currently selected EOL of the pattern S2 or S3, EOL_{n-1} is the previous EOL of one step before, and the EOL_{n+1} is the future EOL of one step after.

In the given manner, the algorithm of the MTA-h scenario is set to segregate the user's EOL selections into three-elements sets of the associated tags. From a cognitive perspective, the given action represents the idea that all close selections in the range of $\{EOL_{n-1} \leq EOL_n \leq EOL_{n+1}\}$ may be interpreted as linked, associated or close-related in terms of their corresponding abstract meanings in comparison to the larger EOL selection intervals. Consequently, every created set contains three composite elements, whose abstract meanings can be characterised as close-related or associated. Upon an EOL_n selection it is more probable that the previously selected EOL_{n-1} and future EOL_{n+1} are more related to each other on a cognitive level of the information processing than the $EOL_{n-7} or EOL_{n-10}$ to the EOL_n. The given assumption is based on the earlier presented characteristics of the cognitive information processing mechanisms with an emphasis on the short-term aspects of the working memory (see section 4.2).

After having initialised the current set of the associated EOL elements, the algorithm is subsequently set to identify within the previously created sets all close-related and associated tags. For the given purposes, the algorithm's function is programmed first to find the element e_1 or/and element e_2 of the current set in all previously created sets (target sets). The function contains a series of key conditions, which stipulate that a target set can have maximum either two elements e_1 or two elements e_2 or maximum one element e_1 and one element e_2. As soon as one of the three conditions is satisfied, the algorithm's function is subsequently set to extract the associated remaining tags from all related target sets, where the elements e_1 and/or e_2 have been found. The extraction process consists of a simple subtraction of the elements e_1 and/or e_2 from the target set by leaving the remaining tag(s) as the result. All remaining tags, which are conventionally different from tags e_1 and e_2 ultimately constitute the associated or close-related newly found tags.

The key objective of the given function is to collect the variations of associated, currently relevant tags, which have been previously selected and analysed by a user during his/her browsing activity. Successively, the given tags, together with the initial tags of the current set e_1 and e_2, are used as background database query inputs for finding the alternative or similar search results, which have not yet been viewed by a user. In the end, the newly found search results, represent the alternative results, which are initially based on the user's current actions in relation to the history of his/her previous browsing activity.

The newly found results represent the last or endmost relevant results in relation to the rest of a list of the stored alternative results. The given algorithm's characteristics allow the MTA-h scenario intuitively and continuously to adapt the list of the collected alternative results to the user's currently prevailing search interests. Therefore, upon the request of alternative results by another Meta-scenario, the algorithm of the requesting scenario always takes into consideration and extracts only the endmost relevant alternative results (see MTA-m scenario 24). The aforementioned function of identifying the associated tags, finding and storing the alternative results constitutes one of the key proactive features of the augmented search engine's mechanism.

Subsequently, after creating the current set of tags, which initially consists only of two elements EOL_{n-1} and EOL_n, the algorithm's next objective is to complete the current set by adding the third final element e_3, or in other words EOL_{n+1}. The given action of closing the current set is necessary as it allows the algorithm to complete the set and to correspondingly place it into a line of the already existing sets of the associated tags.

In order to complete the current set, we add an additional variable, representing a new selected EOL into the algorithm's first part of the Data acquisition. Consequently, the given approach allows us, upon the new rule's iteration, to look for the next selected EOL. As soon as the next selected EOL (EOL_{n+1}) has been detected by the Data acquisition function, the algorithm's second part of actions, dealing with a process of closing the current set is successively activated. The functions of the algorithm's actions simply aim to complete the current set by adding the third EOL element e_3 into the set. After the execution of the given function, the current set has a form of: $\{EOL_{n-1}, EOL_n, EOL_{n+1}\}$. Subsequently, the algorithm's function places the completed set of the associated tags into a line of the already existing sets by assigning to it a corresponding identification number. After the execution of the given function, the current set becomes the last treated set of associated tags (Set_n).

In order to implement the aspect of a process continuity, upon the algorithm's last step, the final fifth part of the algorithm has the objective to execute its corresponding composite functions of the rule regeneration. At the level of the last algorithm's step, depending on various function conditions, the MTA-h scenario is set to regenerate itself with a corresponding type of input parameters. The proactive rule is regenerated either to complete the open set, to initiate the context monitoring of the new S2 or S3 patterns, or simply to continue the context monitoring of the current version of the S2 or S3 patterns.

From a general perspective of the proactive module's mechanism, the MTA-h scenario plays a key role in the underlying process of the search engine's proactive behaviour. An operation of the given Meta-scenario allows us to provide the system with ready-to-use alternative results, which can be requested by another function of the proactive system at any time, if required by a context situation. Upon a state of satisfying the initial function's conditions, the given system scenario allows the algorithm (a) to analyse the history of user's previous actions and (b) to subsequently extract the currently relevant associated and close-related tags. The continuously collected tags are further used by the algorithm for the purposes of finding the alternative, not yet viewed search results. All identified close-related and associated tags, after the background querying of the search engine's database, represent the last and endmost relevant alternative results. The identified alternative results are consequently represented as the closest candidates to the user's currently prevailing search objectives.

The list of all identified alternative results, is stored in a corresponding table of the proactive system's database and is continuously updated with each new iteration of the rule upon conditions that S2 or S3 pattern has been detected. Such approach characteristics allow the MTA-h scenario to continuously adapt the created list of alternative results to the currently prevailing user's search interests. As we will see in the next section, the realisation of the given approach plays an essential role in the extraction of endmost relevant alternative results. Successively, the current algorithm's characteristics allow the corresponding MTA-m scenario to intuitively respond, at any time, to the needs of the currently prevailing context situation.

4.3.3 Threshold-based mechanism of a model monitoring (MTA-m)

The current Meta-scenario, entitled as *MTA-m* is the least complex proactive scenario in terms of its structure and composite functions in comparison to the earlier presented MTA-c and MTA-h scenarios (see Algorithm 24). The main objective of the MTA-m scenario is to provide the user, if necessary, with a list of relevant alternative results. The provided results are extracted from a set of the collected alternative results, which has been initially created by the MTA-h scenario. As we mentioned earlier in the description of MTA-h scenario, the created set of *alternativeResults* represents a list of ready-to-use relevant results, which are always available for any other system's processes. Therefore, upon condition that the *Low relevant results* prevail ($L > H$), the MTA-m scenario is set to provide the user with the last known relevant results, stored in the proactive system's database. In other words, the MTA-m scenario concludes the system's proactive process, first initiated and subsequently carried out by the MTA-c and MTA-h scenarios.

Algorithm 24: MTA-m scenario

Data Acquisition()
boolean threshold ← (Number of completed D-patterns ≥ 5 **and** L-level $\geq 80\%$);

Activation Guards()
return threshold;

Conditions()
return true;

Actions()
selectedResults ← Get last 10 distinct results from *alternativeResults*;

Rule Generation()
if (getActivated) **then**
    createRule displayResults(selectedResults);
    reset number of completed D-patterns to 0;
    createRule MTA-m
else createRule MTA-m;

At the first rule's step of the *Data acquisition*, the algorithm of the MTA-m scenario is set to evaluate the current state of H and L proportional relations. The corresponding function of the algorithm has the objective to detect a set of two conditions. The first condition specifies that a number of the completed D-patterns has to be equal to, or greater than five patterns. Consequently, the given disposition allows us to implement two features, required by the algorithm upon the evaluation of certain context conditions.

The MTA-m scenario is programmed to start considering the results of a statistical data evaluation only after five completed D-patterns. Initially, the given condition is necessary for the purposes of stabilising the ratio of the model's prior probability distribution at the beginning of the user's browsing activity. In other words, if we do not let the model to run the tests, the very first statistical evaluation of one of the behavioural patterns will produce the result of 70% of prevalence either for $P(H)$ or $P(L)$. The main objective of the current condition is therefore to allow the probabilistic model to stabilise its proportional relations between two competing types of the hypothesis. As soon as the ratio of $P(H)$ and $P(L)$ has been stabilised, the algorithm is then set to start to take into account all updated values of the model's prior probabilities.

Furthermore, upon condition that the MTA-m scenario has already displayed a set of alternative results, a period of the five completed D-patterns gives the necessary time to a user to readjust his/her current search querying strategy. After having detected the prevalence of the dissatisfaction model and after having provided a user with a list of alternative results, the algorithm has to wait for a certain period of time, before providing the user with a new list of alternative results. In other words, if at the end of five newly completed patterns of the dissatisfaction model the algorithm detects that the user's search strategy remains unchanged, the MTA-m scenario is then set to provide a different list of the last known relevant results. Consequently, the given approach allows the algorithm to monitor the evolution of user's context actions. Through the continuous and close observation of the user's browsing activity, the MTA-m scenario is ultimately capable to provide an adequate proactive response, which always takes into account all changes in a user's search behaviour.

The second condition of the Data acquisition function specifies that the MTA-m scenario is activated only if the level of $P(L)$ reaches 80% of prevalence. The given context condition represents the function's threshold, which marks the endmost point, indicating the moment of launching the corresponding proactive actions. An implementation of the given condition represents the main technique, allowing the system to determine the exact moments of a potentially required proactive mediation. Therefore, upon the satisfaction of both conditions, that is completing minimum five D-patterns and reaching the threshold of 80% for $P(L)$ the rule's control flow is successively directed to the algorithm's part of *Actions*.

At the fourth step of *Actions*, the algorithm of the MTA-m scenario is simply set to extract from a list of the*alternativeResults* the last ten distinct results. The list *alternativeResults* denotes a record of collected alternative results, created formerly by the MTA-h scenario and stored for a further use in the proactive system database. Each time the function of the MTA-m scenario extracts the results from the list, it always takes the last ten distinct results. If the proactive scenario has already displayed ten alternative results to a user in its previous iterations, upon the next extraction, the algorithm will select again ten last distinct results, but this time omitting those that have been displayed previously.

The design choice of extracting ten last alternative results consequently reflects an idea, stipulating that the given list inherently represents the last known alternative results, which have been collected by the MTA-h scenario upon the ultimate manifestations of the satisfaction model. The given approach allows the system to provide the user with a list of alternative results, which successively might help him in readjusting the currently prevalent search strategy. The MTA-m scenario has the objective to display only ten items at a time in order not to confuse a user with the presented variation of the multiple alternative results. Additionally, the limited number of the presented results allows the system to use the initial list of alternative results more productively, by allocating its finite resources with respect to the potential needs of future context situations.

After the actions-part has been executed, the rule's control flow is successively directed to the algorithm's final step of the rule regeneration. At the given step, the corresponding rule's function is set to delegate the task of displaying the alternative results, by activating a dedicated target scenario, which will conclude the initiated proactive service. After the results have been displayed to a user, the successive function of the given step will reset a number of the completed D-patterns to its initial value of zero. The main objective of the given approach is to provide a user with the needed time, allowing him/her to take the necessary measures for readjusting the currently prevailing search strategy. During a period, when the number of the completed D-patterns is less than five patterns, the rule of the MTA-m scenario will simply regenerate itself for the next iteration, without undertaking any context mediation.

As we may notice, the given third Meta-scenario of the proactive system, represents in itself a relatively less complex algorithmic structure in comparison to the MTA-c and MTA-h scenarios. However, the role of the MTA-m scenario is vital for an accurate functioning of the system's proactive behaviour as it represents an ultimate link between the proactive system and user's environment. The given Meta-scenario is the system's essential function for providing an outward context mediation, which successively concludes an instance of a proactive service, initiated by the MTA-c and MTA-h scenarios.

4.4 Summary

In the current chapter we addressed our earlier introduced, Research Question #3. The given research direction aims to investigate the underlying principles and key approaches, which define the transition of a framework concept from the deterministic towards probabilistic dimension. As we have seen, the system's probabilistic approach consists of several essential elements that conjointly represent its methodological structure. Notably, the mentioned elements are characterised, on the one hand, by an implementation of more complex algorithmic mechanisms of a context data detection and, on the other hand, by an integration of additional context data evaluation techniques.

The first element is mainly represented by the system's algorithmic mechanisms, allowing to detect the various aspects of context's metaphysical characteristics. The given context characteristics are initially defined as instances of the user's hidden cognitive states, manifested during his/her search querying activity. The second element of the implemented probabilistic approach is represented by the system's data evaluation techniques, whose capacity has to be accordingly adapted to the type of the target context data. In other words, the data analytic's techniques have to be able to accurately evaluate the intricate aspects of the user's metaphysical domain.

Our exploratory study, characterised by the implementation of a probabilistic approach, mainly consists of designing, building and testing a prototype version of the user-aware proactive system. The term *user-aware* denotes the system's capabilities, allowing not only to monitor the user's context actions, but furthermore to be able to detect the user's currently prevailing mental characteristics, in form of a cognitive state of satisfaction and dissatisfaction.

The given factor plays a crucial role in choosing the search engine as the system's target environment. The SE environment, besides user's context actions, inherently includes a variation of user's manifested cognitive states. Every time the user encounters the desired high relevant results, through the manifested instances of his context behaviour, the user unintentionally expresses the corresponding mental state of satisfaction. Accordingly, upon an encounter of the unwanted low relevant results, the user unintentionally expresses his mental state of dissatisfaction, which is equally manifested through the instances of user's context behaviour. Therefore, in order to be able to identify, if the currently displayed search results are of high or low relevance, we implement a mechanism, which allows us to detect the related context characteristics of a corresponding abstract domain.

Consequently, after having identified in time, if the current search results are of low relevance, we are able to mediate the ongoing context conditions by proactively providing a user with a list of alternative results. The underlying principles of the proactive mediation consist of the three system processes. The first system process is represented by the MTA-c scenario, whose objective is to continuously monitor the outward context environment for the purposes of detecting the user's currently prevailing cognitive state. The second system process is represented by the MTA-h scenario, whose main objective is to collect the alternative high relevant results, while the user is in his/her satisfaction state. The third system process is represented by the MTA-m scenario, whose main objective is to monitor the proportional relations of high and low relevant results, which correspondingly reflect the user's current cognitive characteristics.

Therefore, upon the detection of low relevant results, through the monitoring of the $P(L)$ level, we know that at the given moment the user's cognitive characteristics are most probably undergoing a substantial emotional shift. In order to prevent the further development of the user's mental dissatisfaction, we provide him/her with the last known high relevant alternative results. In other words, while the $P(H)$ and S-model prevail, we identify and collect the relevant alternative results for the potential future needs. While the $P(L)$ and D-model prevail, we provide a user with the previously collected relevant results at a moment when the proactive mediation is most needed.

The main objective of the present chapter is characterised by a formal description of our second empirical study, which aims to explore and to validate the underlying methodological and theoretical principles of the proactive system's probabilistic framework. The given study objectives represent the consecutive strategic step of our progressive research methodology. In other words, the probabilistic framework represents the successive conceptual evolution of the proactive system's structure, which is based on the fundamental principles of the earlier presented deterministic framework. The final conceptual and methodological basis of the current proactive system prototype is therefore built upon the twofold structure of the deterministic and probabilistic frameworks. The former provides an essential conceptual basis of a proactive rule, proactive scenario and the rule's engine, whereas the latter provides the successive probabilistic techniques of a context data detection and evaluation.

The conjoint application of both theoretical frameworks ultimately allows us to build a prototype version of the proactive system, which is capable to continuously monitor the user's context for the purposes of detection the potential situational problems.

Furthermore, the proactive system is capable to intuitively mediate the target context environment with respect to both, the user's situational needs and user's currently prevailing cognitive states. The twofold methodological structure and the current prototype version of the proactive system, conjointly represent the proof-of-concept, which aims to demonstrate the feasibility and benefit of the applied proactive principles and their functional potentials.

Part III

Results

Chapter 5

Study results and general discussion

The present chapter represents the conclusive part of our work, which aims to consolidate all previously discussed elements of the theoretical and methodological modules. We successively highlight all relevant results, issued from our first empirical study. The experiment has been carried out within the academic environment and has been based on the usability study of the augmented Moodle platform. The main objective of the present chapter is therefore to highlight the key characteristics of the obtained empirical results. The stated approach will consequently allow us to validate our theoretical and methodological frameworks on the initially specified deterministic level of the proactive system structure. Moreover, an additional objective of the current chapter is to successively interrelate in our analytical section all previously introduced concepts, theories and approaches, together with the results of the first *enquiry study* and methodological characteristics of the second *exploratory study*.

It should be noted, that the second experiment is primarily characterised as an *exploratory study*, which does not include the usability investigations nor any quality assessment objectives. The second empirical study aims to demonstrate the *feasibility* of the concept realisation. The given study does not involve the participation of user-subjects, but aims instead to delineate the character and specifications of all applied methodological mechanisms. All issued results from our exploratory study are mainly based on the documented observations, highlighting the methodological characteristics of the chosen approaches, specifications of the system's composite functions and its operative aspects. The current chapter will present therefore only the empirical findings, gained from our first experiment, which aimed to validate the fundamental mechanisms of the proactive system and to concurrently assess their functional impact on a user's overall performance.

In the following sections we proceed in accordance with the aforementioned structure, where we first delineate in section 5.1 all acquired data, which represents the empirical findings, reflecting the functional and usability characteristics of the proactive system. Subsequently, we analyse, correlate and compare all obtained results in our discussion section 5.2. We present in an analytical form the relationships between the initially stipulated study objectives, chosen research methods, applied theoretical and methodological techniques and the subsequently acquired empirical data. Additionally, we analyse the findings with regard to the chosen approaches and methods, followed by a further discussion of the results' analysis in the framework of our theoretical module.

5.1 Experiment's results

In the current section, we present the empirical results, which have been acquired throughout the first empirical study. The given experiment aims to validate our initial theoretical and methodological framework on a fundamental deterministic level. As described in section 3.2.7, the current enquiry study has taken place at the University of Luxembourg at the faculty of Computer Science and Communications for a period of one academic semester. The experiment consisted of using the University's electronic platform *Moodle* for the purposes of testing the newly designed mechanisms of the proactive system. The corresponding *proactive module* has been implemented as a system plugin, which allowed us to augment the Moodle's basic functions with a proactive type of system behaviour.

The main objectives of the present study are characterised by two enquiry categories. On the one hand, we investigate all associated system-related aspects of the proactive system functioning, on the other hand, we examine certain inherent and ensuing aspects of the learning-related domain. The *system-related category* aims to study several system-based factors, including the general quality of the system's operation and software reliability, as well as the system's visual characteristics, represented in form of the implemented graphical user-interface. Subsequently, the mentioned category is divided in two subcategories of the *User-computer interactions* and *System statistics*. On the other hand, the *learning-related category* aims to investigate the aspects of the user's e-learning performance, which include but are not limited to such phenomena as the practice of an e-learning activity, cooperative and collaborative learning, students' general performance, motivation and so on.

Course	Assignment	Enrolled students total	control	study	Submitted assignments total	%	control	%	study	%	diff.
Algorithmics 2	Assignment 1	18	9	9	8	44.4%	4	44.4%	4	44.4%	
Algorithmics 2	Assignment 2	18	9	9	9	50.0%	4	44.4%	5	55.6%	
Algorithmics 2	Assignment 3	18	9	9	10	55.6%	4	44.4%	6	66.7%	
Algorithmics 2	Assignment 4	18	9	9	9	50.0%	2	22.2%	7	77.8%	
Algorithmics 2		18	9	9		50.0%		38.9%		61.1%	22.2%
Probabilities	Assignment 1	41	20	21	22	53.7%	9	45.0%	13	61.9%	
Probabilities	Assignment 2	41	20	21	30	73.2%	12	60.0%	18	85.7%	
Probabilities	Assignment 3	41	20	21	8	19.5%	3	15.0%	5	23.8%	
Probabilities		41	20	21		48.8%		40.0%		57.1%	17.1%
TOTALS		59	29	30		49.5%		39.4%		59.4%	20.0%
Probabilities (without Assignment 3)						63.4%		50.0%		77.5%	27.5%
TOTALS (without Assignment 3)						54.5%		43.4%		65.3%	21.9%

TABLE 5.1: Assignments' submission level

5.1.1 Learning-related category

The key objective of the present category is to detect and to investigate a variation of potential cognitive effects on a user's performance in relation to his/her learning activity. The given target characteristics are consequently stipulated by the conditional possibility that the user's e-learning performance may be influenced by the implemented proactive features either in a positive or a negative manner. In other words, the key objective of the learning-related category is to specify and to define the second aspect of the determinist approach, which at this point, aims to detect and to identify a likelihood of either positive, negative, or static effects, caused by the implemented proactive features. The given category is successively divided into several subcategories, which represent individually two different types of the experiment's findings.

Assignments submission

The objective of each created proactive scenario is to reflect the underlying cognitive aspects of the students' e-learning behaviour, which is consequently conditioned by the specifics of the students' general learning activity. In other words, every type of the system's proactive behaviour, including each notification message, possesses its own predefined objectives. With regard to a type of the chosen target environment, the main orientation for all designed instances of the system's proactive behaviour is to stimulate the student's learning performance during the process of an assignment accomplishment.

Correspondingly, the collected statistical data aims to highlight the results of the aforementioned activity. In other words the results, issued from the first experiment, have

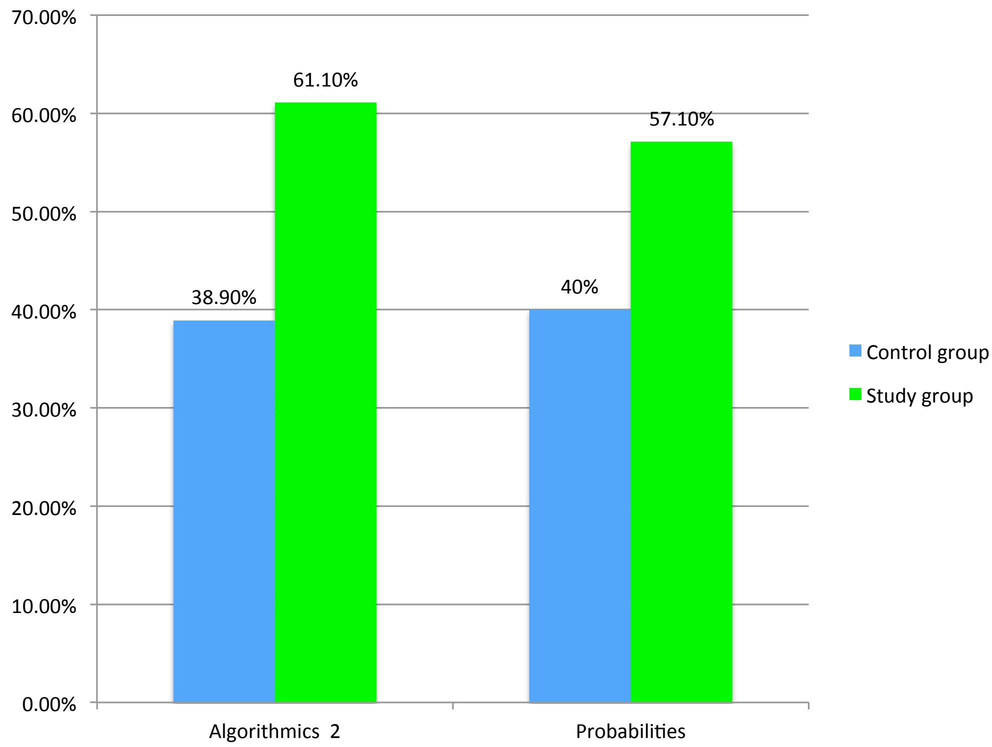

FIGURE 5.1: Assignment submission variance for the courses *Algorithmics 2* and *Probabilities*

the ultimate objective to reflect the achieved level of a students' academic activity stimulation, initiated by the proactive system. The results, characterising the given category are presented in Table 5.1.

During the current empirical study, we have collected the statistical data only from two corresponding courses, at the faculty of Computer Science and Communications, including *Algorithmics 2* and *Probabilities*. All gathered data represents the students' activity, performed during one academic semester, which in total contains seven course assignments. Consequently, we have eighteen students, enrolled in the course *Algorithmics 2*, where nine students belong to the *control group* and other nine respectively to the *study group*. For the second course of *Probabilities*, we have forty-one enrolled students in total, where twenty students constitute the control group, and other twenty-one constitute the study group.

We begin the description of all associated experiment's results by presenting first the level of assignments' submissions for each given group, divided respectively by our two aforementioned courses. Consequently, we have 38.9% of total submitted assignment tasks for the course *Algorithmics 2*, performed by the students of the control group. Correspondingly, we have 61.1% of total submitted assignment tasks for the same course *Algorithmics 2*, performed by the students of the study group. As we may notice, the

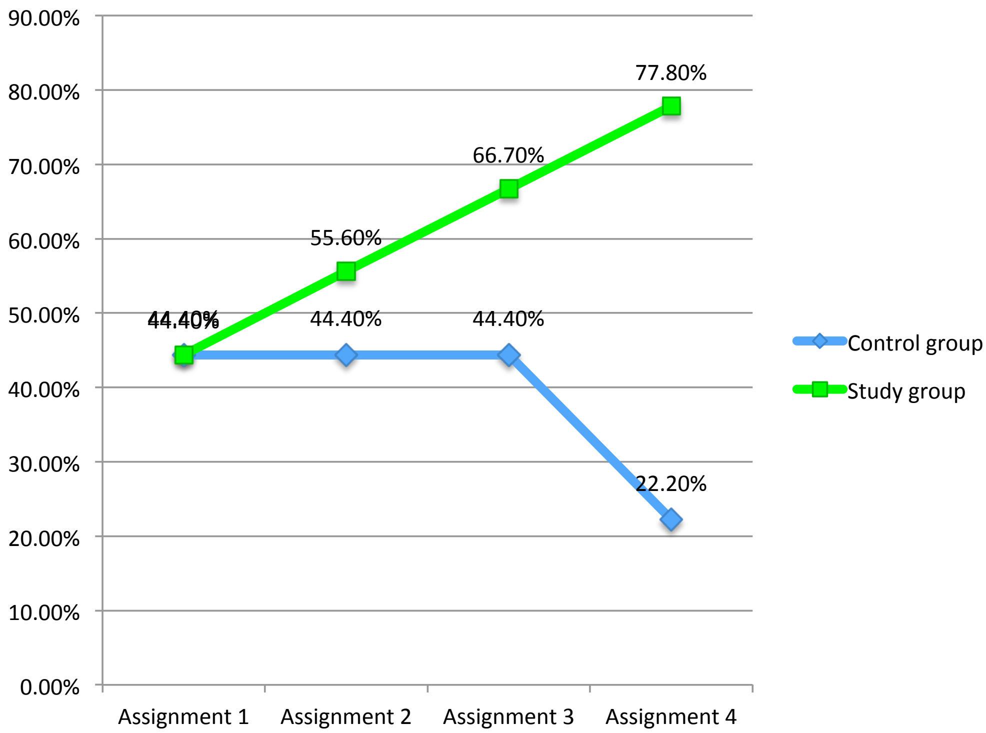

FIGURE 5.2: Submission variance for each assignment in the course *Algorithmics 2*

according difference in overall assignments' submission level is represented by 22.2 percentage points of advantage for the study group. The given difference in the assignment submission level is depicted in Figure 5.1.

If we take a look on all results, related to the course of *Algorithmics 2* individually, we may notice that for the first assignment the number of the submitted tasks for each group is uniform, that is equal to 44.4% for both groups. For the second and third assignments, the number of the submitted tasks starts to vary with a slightly increasing advantage for the study group, including 55.6% against 44.4% for the second assignment, and 66.7% against 44.4% for the third assignment. The last assignment of the given course marks the biggest attained interval in the task submission level with the evident 77.8% for the study group against the respective 22.2% for the control group. The variations of the submission process for all mentioned assignments of the course *Algorithmics 2* are depicted in Figure 5.2.

In relation to our second study course of *Probabilities*, we have 40% of total submitted assignment tasks, performed by the students of the control group. Consequently, we have 57.1% of total submitted assignment tasks for the given course, attained by the students of the study group. Similarly to the previous study course, we may notice the according difference in the overall assignments' submission level, which is correspondingly represented by 17.1 percentage points of advantage for the study group. The given

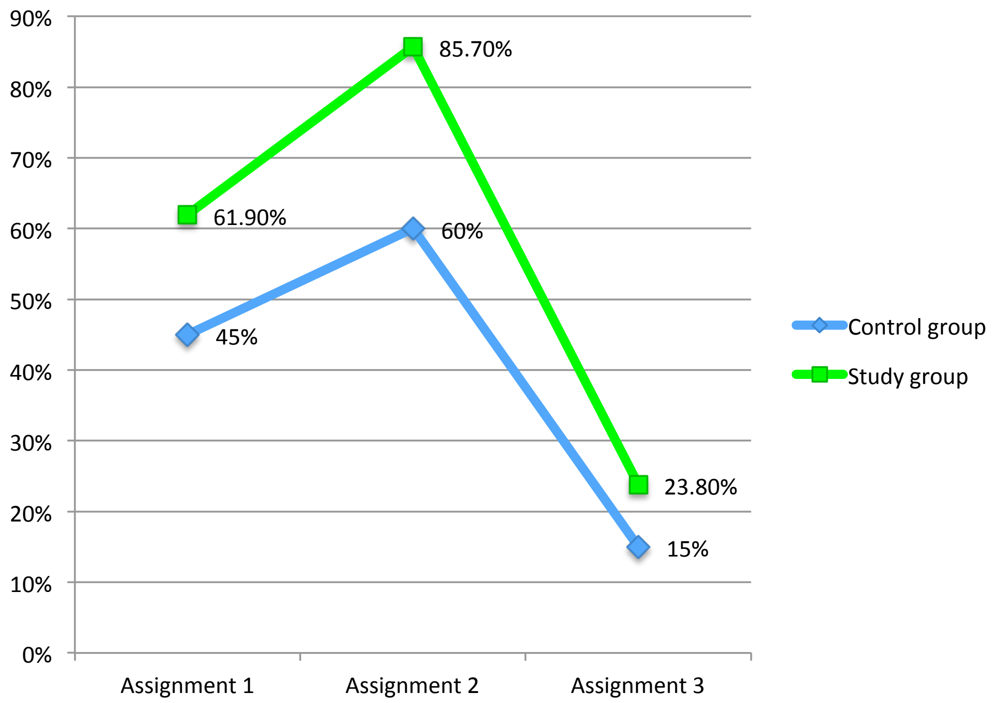

FIGURE 5.3: Submission variance for each assignment in the course *Probabilities*

difference in the assignment submission level is depicted in Figure 5.1.

We further delineate all results variations, pertinent to each assignment of the course *Probabilities*. Consequently, from the first look on individual assignments, we may notice that in comparison to the *Algorithmics 2* the course *Probabilities* does not contain any uniform data. By analysing sequentially all course's assignments, we may notice that for the first assignment submission the variation in the final results is represented by 61.9% for the study group against 45% for the control group. Subsequently, for the second assignment the percent of the submission level for the study group keeps to be prevailing with 85.7% against the respective 60% for the control group. Ultimately, for the last assignment the corresponding submission level of the students' accomplished tasks is represented by 23.8% for the study group against 15% for the control group. The variations of a submission process for all mentioned assignments of the course *Probabilities* are depicted in Figure 5.3.

As we may notice, the Table 5.1, indicating the level of all assignments' submissions contains furthermore several additional lines, which depict the alternative results for the course of *Probabilities*. The table contains a line, which delineates the total of assignment submissions for the given course without the assignment 3. Such decision of recalculating the results with only the first two assignments is essentially stipulated by a significantly low number of submissions of the given assignment by both groups, with corresponding 15% for the control group against 23.8% for the study group. We hypothesise that the

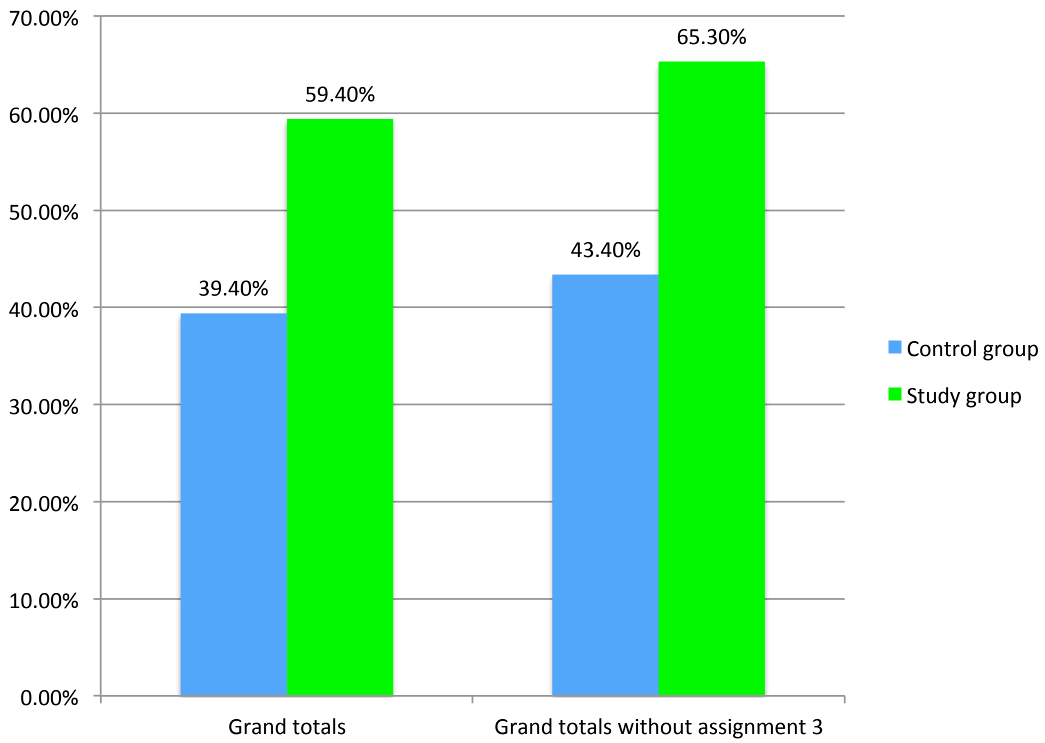

FIGURE 5.4: Submission variance for all courses in a form of grand totals

given characteristics of the third exercise are probably due to the students assignment submissions off Moodle platform. Consequently, after recalculating the data, we have 50% of assignment submissions for the control group against the respective 77.5% for the study group. Ultimately, the study group attains 27.5 percentage points of advantage over the control group in the category of all assignment submissions in the course of *Probabilities* without assignment 3.

In relation to the grand totals for both courses, we have 39.4% of all submitted assignments for the control group, and respectively 59.4% of all submitted assignments for the study group. Consequently, for the grand totals of both courses the study group attains 20 percentage points of advantage over the control group in all submitted assignments of the given category. Respectively, the grand totals without assignment 3 of the *Probabilities* course are represented as 43.4% of all submitted assignments for the control group against 65.3% for the study group. Consequently, for the grand totals of both courses without assignment 3 the study group attains 21.9 percentage points of advantage over the control group in all submitted assignments of the given category. The variations of a submission process for all mentioned courses in form of grand totals are depicted in Figure 5.4.

Students' final grades

As we have noted, the data from Table 5.1 represents only one category of experiment's results. The given data has the key objective to highlight the underlying characteristics and important traits of the students' assignment submission process. The presented results' data consequently reflects only one side of the learning-related category of our empirical study. In order to take a look on a developing situation from a different angle, we decided to analyse the students' academic performance for both courses in form of a comparative analysis of students' final grades. The given approach allows us to view from a general perspective the level of the proactive system's potential impact on students' overall academic performance in form of the final grades, which subsequently represent the global scale character of our empirical data analysis.

We categorise the issued experiment's data respectively by a course and group type of the participating students. Therefore, the structure of our results' data, presented in Table 5.2 reflects the students' academic performance for both, control group and study group, related to our earlier mentioned courses of *Algorithmics 2* and *Probabilities*. The given results' data delineates the total number of enrolled students for both courses and the respective number of students, who passed or succeeded in the final exams at the end of a semester.

We start the description of our results' data by presenting first the associated percentage of the passed students for each given group by a corresponding course. In relation to the number of students, who passed the final exams for the study course of *Algorithmics 2*, we have 0% of the succeeded students of the control group against 33.3% of succeeded students of the related study group. Essentially, the given number of 0% indicates that none of the students of the control group has passed the final exams at the end of an academic semester. Correspondingly, in relation to a number of students, who passed the final exams for the second study course of *Probabilities*, we have 70% of succeeded students of the control group against the respective 71.4% of succeeded students of the study group.

Ultimately, the grand totals, indicating a percentage of students of each group, who passed the final exams at the end of the academic semester are represented as 48.3% for

Course	Enrolled students			Students who passed the final exams						diff.
	total	control	study	total	%	control	%	study	%	
Algorithmics 2	18	9	9	3	16.7%	0	0.0%	3	33.3%	
Probabilities	41	20	21	29	70.7%	14	70.0%	15	71.4%	
TOTALS	59	29	30	32	54.2%	14	48.3%	18	60.0%	11.7%

TABLE 5.2: Variations of students' final grades

the control group against the corresponding 60% for the study group. As we may notice from the results' data, the according difference in the overall process of passing the final exams is equivalently represented by 11.7 percentage points of advantage for the study group. Consequently, the variations of students' final grades for both courses in form of grand totals are depicted in Figure 5.5.

The given data delineates the characteristics of the experiment's results, which aim to highlight the underlying traits of the students' overall academic performance within the period of one study semester. All empirical data, presented above, consequently depicts the fundamental properties of our experiment's results on the first-mentioned level of the *learning-related* aspects. Subsequently, in the following paragraphs we present a data, which characterises the experiment's results on the second-mentioned level of the *software-related* aspects. The information of the given level aims to successively interrelate the data from both results' categories. In the upcoming paragraphs, we will see how an overall level of the students' academic performance may be potentially related to the specific usage of the augmented Moodle platform.

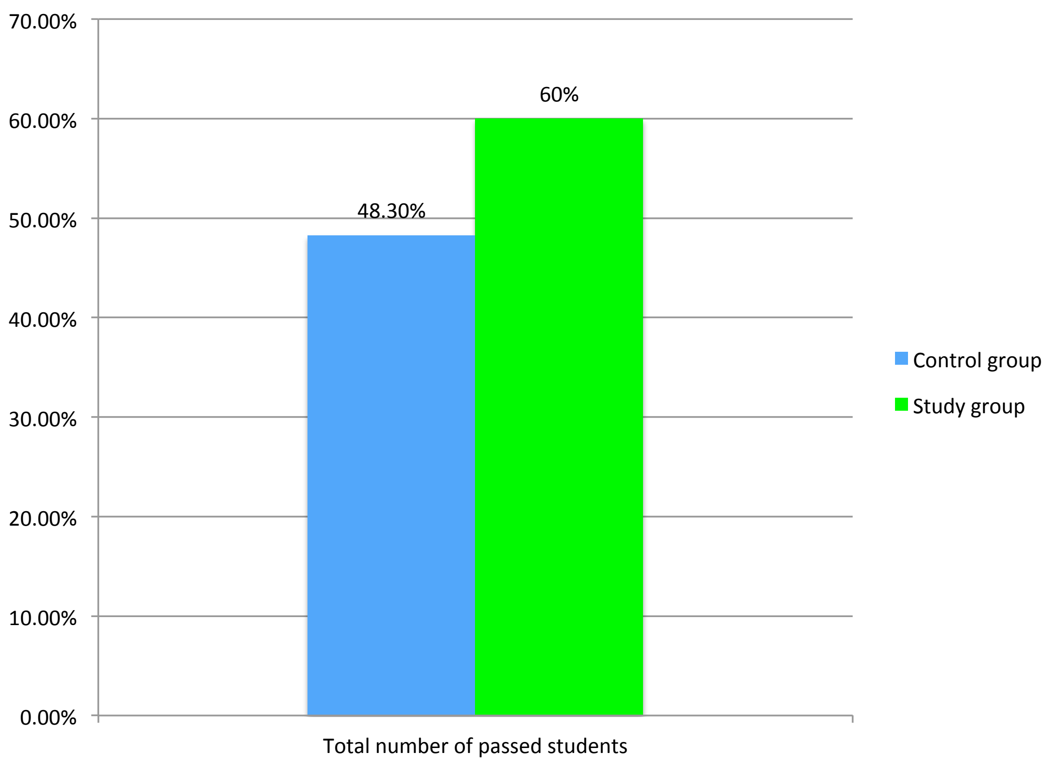

FIGURE 5.5: Variations of students' final grades for both courses in a form of grand totals

5.1.2 System-related category

In the given subsection, we present all related results' data, which is initially associated with a subcategory of the user-computer interactions. Consequently, the key objective of the *software-related* category is to depict a potential relationship between the earlier presented *learning-related* aspects and characteristics of the user's direct interactions with the system. It should be noted that the students' perceptions of a corresponding graphical user interface as well as the subsequent interactions with its compound elements play an important role in the overall process of the proactive system optimisation. Therefore, the current category of the software-related aspects is conventionally divided in two subcategories, dealing respectively with the user-computer interactions and system's internal processes.

User-computer interactions

In order to create a direct link between the proactive system and its target environment, we implement a graphical user interface, which allows us to provide the necessary informational exchange between the system and user. Furthermore, the implemented elements of the user interface allow the students to interact with the proactive system through a set of simple commands, including opening or deleting a message and opening the list of messages. The characteristics of the integrated *Messaging block* equally allow a user to see the headlines of the unread messages. For the given purposes, we have collected the statistical data, which highlights the specifics of the user's interactions with our implemented Messaging block inside of the Moodle's webpage.

All gathered statistical data shows that during one academic semester, there have been 251 explicit interactions with our Messaging block, including the browsing actions of opening a message and deleting it, either prior to viewing or after it. Additionally, the system has detected 224 messages that show no explicit user-computer interactions, meaning a user neither deleted nor opened the displayed message. The given data may indicate however that a user has read at least the heading of a message as it is always visible to a user without any direct actions.

As we have delineated in our earlier sections, additionally to the statistical data collection, we create an electronic survey or questionnaire, which initially aims to serve as a complementary data source for identifying the users' views and opinions, regarding new proactive functions of the Moodle platform. Furthermore, all collected survey data aims to highlight the user's perceptions and opinions, related to the underlying characteristics of visual and interactive aspects of the Moodle's implemented user interface elements.

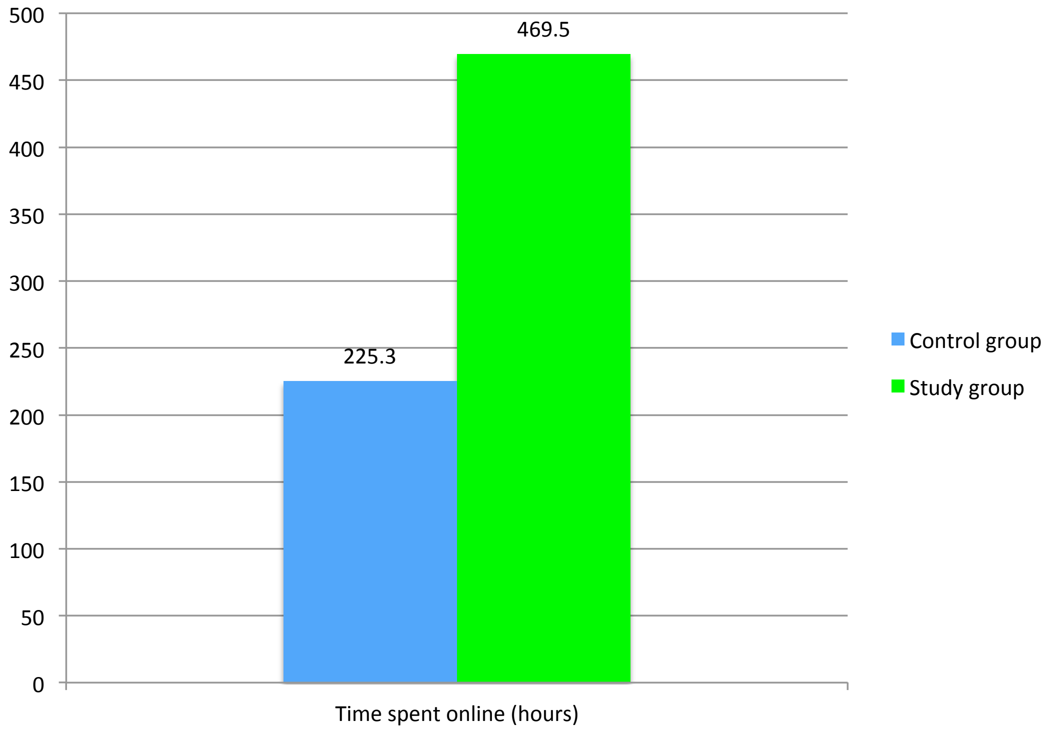

FIGURE 5.6: Variations of students' total time of stay within the Moodle's environment for both groups

However, due to the low rate of the received survey answers, we successively decided not to use the questionnaire data for any quantitative analysis, but instead to employ it for the later qualitative analysis upon the next system's optimisation steps.

Ultimately, the last data characteristics of user-computer interactions are related to the aspect of online time variations, which initially indicate the corresponding users' total time of stay on the Moodle platform for both groups of students. The following data specifies the total online time that has been spent by a user within the associated Moodle's webpages during one academic semester. The given data shows the related time values respectively for the control and study groups. According to the collected statistical data, we have 225.3 hours of online browsing, spent by the control group against 469.5 hours of online browsing, spent by the study group. Consequently, the time difference between two groups is represented by 244.2 hours of the additional online browsing, spent by the study group within the Moodle's environment in comparison to the corresponding time value of the control group. The variations of the mentioned time values for both groups are depicted in Figure 5.6.

System statistics

The statistical data, described in the following paragraphs represents the system's overall performance during the given period of one academic semester. The results' data highlights various sides of the system's internal processes, which initially are characterised as the underlying mechanisms of the system's implemented proactive behaviour. The data, presented below, highlights the number of system's iterations, number of the executed proactive scenarios, number of the communicated proactive notifications and the average time value of a single iteration.

Thus, the collected data shows that for the running time of one academic semester the proactive system performed 772050 consecutive iterations, where it executed 6151617 Meta-scenarios and respectively 1326 Target-scenarios. Consequently, 475 messages have been sent to the users by the proactive system during a total period of one academic semester. Additionally, in relation to the characteristics of its internal performance, the system's single iteration period took in average 20 milliseconds during which, between 4 and 15 proactive rules have been executed. The given period of 20 milliseconds equally includes the access of the proactive system into the Moodle's database.

5.1.3 Section summary

As we have seen, the given section deals with the results of our first empirical study, which highlights two different types of the collected results, including learning-related and software-related categories. The former category of the learning-related aspects describes the characteristics of the students' overall academic performance, which is essentially based on two underlying factors, that is an assignment task accomplishment and quality of the students' final grades. Both subcategories aim to highlight the characteristics of the students' general learning performance, which has been partially affected during one academic semester by the stipulating factors of a proactive system behaviour.

Additionally, we highlighted the experiment's results, which characterise the second category of the system-related aspects. In the designated part we described two associated factors, dealing respectively with the user-computer interactions and specific, system-related functional aspects. In combination, both subcategories demonstrated several underlying links between the system's functional characteristics and user's accounted interactive feedbacks.

5.2 General discussion

In the current section we present, in an analytical and comparative manner, the main orientations and details of our understanding and interpretation of the earlier delineated experiment' results. We present the underlying relationships between the initial objectives, experiments' findings, methodology and the results' analysis. Therefore, the key objective of the present section is to unify in a discursive form the main points of the earlier presented chapters by highlighting the main relational factors of the experiments' results with regard to our initially stipulated study objectives. In the upcoming subsections we discuss the significance of the results by highlighting their relationship to the chosen methods and characteristics of the described theoretical background. Ultimately, we take a look on the experiments' results from the perspective of their wider meanings and implications for the existing knowledge.

5.2.1 Findings in relation to the thesis' methods and approaches

As we have specified in our previous chapters, the main orientations and objectives of the given study are characterised by a process of elaborating a stable conceptual framework, based on the twofold structure of the deterministic and probabilistic principles. Correspondingly, the first experiment and its successive empirical outcomes, addressed the first mentioned methodological aspect or, in other words, the deterministic side of the proactive system framework. Therefore, all issued experiment's results consequently highlight the underlying aspects of the proactive system's concept on the given fundamental level. After a thorough analysis of the experiment's results, we have identified two main factors, which ultimately represent the findings' main implications.

Implications of the first empirical study

Our first category of results' interpretation, which specifies the findings' significance in terms of the chosen methods and approaches, characterises the currently employed proactive system concept as a functional and effective framework structure. In other words, the initial results of the first experiment demonstrate that the proactive system may effectively operate on the following underlying principles:

- Predetermined structure of a rule's policy,
- Sustained engine iteration,
- Action task delegation,
- Proactive scenarios' mechanism of a target context monitoring [3, 5, 10].

As we have seen in the earlier presented results' sections, our proactive system is capable to effectively sustain on a continuous basis the iteration of its main functions, that is an activation and execution of all associated proactive rules. During one academic semester, the proactive system has been capable to continuously perform 772050 iterations, where it executed 6151617 Meta-scenarios and 1326 Target-scenarios, which have been used as the system's main proactive mechanism.

The applied principles of a task delegation, by means of implemented proactive scenarios, have demonstrated that the chosen approach is an effective mechanism, allowing to continuously monitor the various aspects of a user's context environment. As we have seen in the performed experiments, by applying the principles of proactive scenarios, we are able to decontextualise certain aspects of an outward target environment [52]. The given approach is implemented through modelling and simulating the individual representations of particular context events. Consequently, the modelled instances of various context events allow the system, upon the engine's iteration, to detect the individual deviations of the specified context situations.

Therefore, the multitude of the designed proactive scenarios, collectively represents an abstraction model of a particular context setting, treated by our system. The approach of the context fragmentation, through modelled proactive scenarios, consequently allows us to create an actual representation of desired events for the purposes of their successive proactive mediation. As we have seen in our both experiments, the given concept-approach represents a functional model, allowing to effectively monitor an associated context environment.

All aforementioned conceptual characteristics, applied within the current system framework and subsequently tested during our first empirical study, allow us to validate the designated proactive system's principles on a fundamental level of the deterministic approach. In other words, our first empirical study demonstrates that the applied deterministic mechanisms of the current system framework allow us to effectively implement the corresponding basic principles of the proactive computing paradigm.

The empirical data, issued from our first experiment, additionally indicates that the currently applied proactive system concepts are capable to constructively reflect the original characteristics of proactive computing principles, initially proposed by Dr. David L. Tennenhouse [1, 2]. In other words, the system's functional characteristics of a task delegation and its compound operative orientations towards the current needs of a user, represent the proactive computing premises, mentioned by Tennenhouse in his prospective visions of a new paradigm.

In relation to the specified principles of the proactive computing and according to the earlier presented results, we may therefore state that our system always works on behalf of a user with the underlying objective to provide the user with the necessary proactive assistance, where such involvement is needed. On the other hand, our proactive system always functions at a higher operational frequency than a user could possibly sustain. The given disposition is mainly represented by the continuous information exchange between various proactive scenarios and their associated composite functions. As we have mentioned in our theoretical chapter, the proactive system's operational frequencies are much faster, which puts automatically a human agent outside of the interactional loop. In this perspective, the computing processes of our proactive system have intrinsically faster response rate to an external stimuli in comparison to a situation where the human decision making is involved (see section 2.1.1). Collectively, all aforementioned functional aspects of our system successively cover the main principles of the proactive computing paradigm, initially proposed by David L. Tennenhouse.

The current category of results' interpretation, which specifies the findings' significance in terms of the chosen methods and approaches, additionally indicates the need for designing more elaborate context monitoring mechanisms for our proactive system. In other words, the experiment's findings stipulate that the currently implemented event-monitoring techniques of the proactive system require more intricate and evolved mechanisms of an outward context analysis. The new sensory mechanisms have to be sensitive enough in order to be always able to detect the hidden metaphysical aspects of the user's cognitive characteristics.

The given approach disposition is based respectively on the quantitative and qualitative results. The quantitative data is mainly represented by the collected statistics, which highlight the character of the user's interactions with our proactive messaging block. The qualitative data is initially obtained from the organised electronic survey. Due to the low rate of the questionnaire answers, we decided not to use the collected data for any statistical calculations but instead to employ it for a further qualitative analysis.

The gathered survey data consequently indicates several proactive system's behavioural aspects. One of the identified aspects, highlighted by the collected questionnaire data, emphasises a particular context situation where the proactive notification messages have been sent in a *non-intuitive* manner, and thus negatively affected the user's emotional state. The term *non-intuitive* may be successively defined by the context condition where proactive notifications arrive to a user at an undesirable time and with an irritating high frequency. As highlighted by the researchers from the Carnegie Mellon University in their visionary paper about pervasive computing, *Proactivity* in itself contains a potential risk of annoyance, which may obstruct the fulfilment of the initial goal of invisibility [20].

Unless it is carefully designed and accurately optimised, proactive computing may have a negative impact on a user's emotional state. Consequently, the system's new context monitoring techniques have to possess more intuitive sensory mechanisms, which will allow us to decrease the system's obtrusive behaviour.

Conventionally, the system's context monitoring capabilities are represented by a variation of the implemented proactive scenarios. Therefore, the highlighted above empirical finding stipulates that for the purposes of more intuitive context monitoring techniques, we first of all need to elaborate the compound structure of our proactive scenarios. The new proactive scenarios have to be more intuitive not only in relation to a simple context event detection, but equally in relation to the sensitivity towards user's prevailing cognitive characteristics. Thus, we may assert that the user's underlying cognitive characteristics, manifested within the target context environment are the composite elements of an evolving context situation. However, the given level of context details is usually not accessible through the ordinary quantitative techniques of an event monitoring. The new proactive scenarios, must therefore be able to address *qualitatively* the process of gathering and analysing the associated instances of a context data.

The main point of the currently discussed empirical findings, stipulates that the initial basic structure of the system's monitoring techniques is characterised as functional and effective. However, for the purposes of a more intuitive context event monitoring, the aforementioned fundamental mechanisms of the data interpretation require a more elaborate adjustment with regard to the hidden aspects of the user's cognitive states.

Consequently, the highlighted characteristics of the first experiment's results ultimately create the basis and corresponding guiding principles for a succeeding extension of the proactive system's initial features. The system evolution, in terms of its new designed features and their intrinsic qualities, successively implies the need of elaborating and implementing the new approaches and techniques. Correspondingly, all new implemented methods and concepts have to be subsequently tested and validated, in terms of their coherence and effectiveness. Therefore, the main characteristics of our research state at the moment of the first results' analysis, ultimately lead us towards the design and elaboration of the successive second set of empirical studies.

The underlying basis for the second experiment is conditioned by the analysis of the results, issued from our first empirical study. The motivation to design more adapted and intuitive proactive scenarios, capable to accurately identify the exact moments for a proactive mediation, consequently prompts us to set up the second empirical study. The main objectives of the second experiment are to elaborate the new according methods, techniques and approaches, which will accurately respond to the earlier identified methodological needs. Correspondingly, the second experiment and its successive empirical outcomes address the second mentioned methodological aspect of our study or, in other words, the probabilistic side of the proactive system framework.

Implications of the second empirical study

It should be noted, that the second experiment is primarily characterised as an exploratory study, which does not include the participation of user-subjects nor any quality assessment objectives. Initially, the second empirical study aims to demonstrate the *feasibility* of the concept realisation. Consequently, all issued results are mainly based on the documented observations, related to the concept's methodological characteristics, functions' implementation specifications, and the system's operative aspects.

Our current category of results' interpretation, which specifies the findings' significance in terms of the chosen methods and approaches, collectively defines all applied methods and approaches as an effective methodological strategy for the required concept realisation. The study results indicate that a more complex structure of the second-wave proactive scenarios may be successfully based on the principles of the probabilistic approach and cognitive modelling methodology [11, 35, 58].

Collectively, the techniques of both approaches incorporate more accurate and intricate mechanisms of a context data evaluation. The cognitive modelling approach allows us to tune the mechanisms of the proactive system to the hidden aspects of the user's cognitive characteristics. The application of the mentioned approach allows the system to access deeper levels of corresponding context settings. On the other hand, the implemented probabilistic methods allow the system to accurately estimate the properties of the detected context data and to specify the most relevant time for corresponding proactive actions. Furthermore, the cognitive modelling methodology inherently incorporates the interdisciplinary aspect, and thus allows us to extend the functional limits of the proactive system and to increase an accurateness level of its context monitoring processes.

Consequently, we may highlight the threefold benefit, resulted from implementing the cognitive modelling approach. (1) First of all, an application of the given methodology provides our research with an interdisciplinary aspect, which consequently allows us to view the underlying mechanisms of the proactive system from a multitude of different new perspectives. The given factor allows us to go beyond the "standard" orientations of viewing the system's composite features and their defining characteristics. (2) The application of the cognitive modelling approach, subsequently provides our system with a possibility to access deeper layers of the associated context settings. The given disposition allows the system to detect an entirely new range of information, and thus to acquire more extensive and finer context details. (3) Ultimately, the application of the cognitive modelling methodology allows us to put into place the chosen probabilistic approach, which is initially applied for the purposes of evaluating the proportional relations between two cognitive models.

Our current category of results' interpretation additionally specifies the necessary conditions for an effective functioning of the probabilistic framework. The observations of the system's functional aspects indicate that the applied probabilistic and cognitive modelling techniques consequently necessitate a thorough parameter optimisation and overall models' refinement. In other words, the identified empirical findings highlight an importance of specifying more accurately the composite function's parameters of the probabilistic model as well as an importance of adjusting the modelled instances of a user's behaviour.

The applied probabilistic model requires the parameters' adjustment for a more accurate evaluation of the data between the D-model and S-model. On the other hand the modelled behavioural patterns necessitate a more rigorous refinement of their composite sequences of the allocated statistical data. The given models' optimisation is a requisite condition for the overall system refinement as all modelled instances of the user's behavioural patterns initially touch an intricate level of human cognitive aspects. For an effective models' optimisation, we have to thoroughly generalise their compound specifications in order to allow the models to cover a multitude of human cognitive variations. In other words, more deeply we try to penetrate into context details, more refinement and adjustment our system needs for an accurate detection of a target context data.

Consequently, in order to attune the proactive system in the mentioned way, an additional set of experiments is required. The new empirical study will have the key objective to investigate the most prevalent characteristics of a user's browsing behaviour in relation to two cognitive states of the mental satisfaction and dissatisfaction. In order to effectively identify all pertinent and composite specifications for both cognitive states, the experiment's settings will have to include the user-subjects, allowing us to create the necessary study and control groups. Subsequently, we will have to record and analyse all users' manifested behavioural patterns, performed during an online search activity. All collected instances of the user's behaviour will have to be generalised and sorted according to the quality of their expressive characteristics. The new potential experiments are consequently attributed to future research efforts.

5.2.2 Findings in relation to the theoretical background

Consequently, after having examined the first relationship between the chosen methods and obtained results, we now pass to the discussion of results' interrelations with the established theoretical background. The given approach will allow us to view the findings' implications from a different and more general perspective. The previously discussed results' implications allowed us to identify the validity and effectiveness of the chosen methods and approaches in relation to the project's initial objectives and characteristics of the obtained results. In the following discussion, we successively analyse the results' implications in relation to our initial objectives and elaborated theoretical basis of the current study. The given approach will equally allow us to assess the validity and effectiveness of the chosen methodological techniques.

Implications of the first empirical study

Consequently, in the scope of the current perspective, the obtained results emphasise the following three derived implications. (1) According to the results of the first experiment, the concept of the proactive computing, as it was defined by David Tennenhouse, has the positive effect on a quality and effectiveness of the user's general performance. (2) According to the collected statistics and results of the system's general performance, we may state that it is technically possible to implement the concept of proactivity on a basis of the chosen methods and approaches. (3) Ultimately, in order to move beyond the predefined deterministic structure of the proactive system, and thus to refine its monitoring and mediating processes, we need to qualitatively extend its current conceptual and methodological framework by adapting new theories, methods and approaches. The collected quantitative and qualitative empirical data has distinctly demonstrated an underlying necessity of integrating the expertise of the corresponding cognitive science domain.

Results' implication #1. Our first-mentioned results' implication characterises the quality and character of the proactive computing potentials. According to the earlier presented results of the first experiment, the students' general reaction on the implemented proactive features is primarily defined as positive. The given statement is based on the acquired evidence, represented by the empirical data in Tables 5.1 and 5.2. From a general perspective, the collected statistics specify the quality and effectiveness of the students' overall performance within the academic environment through the partial mediation of the e-learning assisting platform [52]. The augmented Moodle functionality, by proactive type of behaviour, consequently allowed the students of the study group to overreach the students of the control group in several learning-related categories. The given results represent our main argument, characterising the quality impact of the implemented proactive features.

Results' implication #2. The second results' implication initially characterises the feasibility conditions for the implemented proactive features. On the basis of the chosen approaches and issued experiment's results, we may ultimately state that a realisation of the concept into a viable real implementation is possible both, on the technical and empirical levels. In our opinion, the proposed structure of the proactive engine and concept characteristics of the proactive scenarios, collectively provide an effective technical framework for realising and implementing the idea of the proactive computing. Among the proactive system's various characteristics, several may be highlighted as the most essential. The given list includes:

(a) the system's generic structure,

(b) the system's adaptability in relation to new target environments, and

(c) the system's modular framework, which allows us to extend its initial features' stock through a multitude of new proactive scenarios.

The stable operation of the proactive system, may consequently serve as a stable ground, or a proof of concept for all further expansions and developments of the proposed framework.

Results' implication #3. Our third and last results' implication of the first experiment aims to highlight the quality of the initially applied methodological framework and the need for a further conceptual evolution. The analysis of the results suggests a gradual shift of the initial deterministic framework towards a prospective probabilistic approach. The given methodological orientation is stipulated by the composite properties of the system's deterministic mechanisms, which consequently require more elaborate mechanisms for increasing the accurateness level of context monitoring processes. The initial techniques of the deterministic approach allow the system to detect only the elementary superficial types of context events, whereas the techniques of the probabilistic approach, together with the cognitive modelling methodology, allow the system to analyse a context with a more intuitive and perceptive orientation. For the purposes of updating the initial conceptual framework, we elaborate and implement more complex proactive scenarios, which consequently integrate the essential aspects of the probabilistic approach. Upon an adaptation of the given methodological directives, we keep our deterministic framework as a fundamental basis for the successively implemented, additional probabilistic techniques.

Consequently, in order to achieve the higher level of intuitiveness and system's perceptive qualities, we need more than a unique implementation of the probabilistic methods. According to the character of the obtained quantitative and quantitative results, we have subsequently identified a requisite condition, stipulating the necessity to include in our research a more extensive theoretical input from the corresponding cognitive science domain. In other words, in order to endow the system with a more perceptive and intuitive functional capacity, enabling it to capture the deeper context layers of the user's cognitive characteristics, we need to include in our research the theoretical and empirical knowledge of the applied cognitive psychology. In order to be able to detect the corresponding cognitive states of a user, we need to adapt our methodology according.

The given empirical finding plays an important role in our study as it emphasises the underlying link between the proactive computing paradigm and cognitive science research. The stipulated results' implication of the first empirical study rationalises the necessity of designing and building a thorough multidisciplinary methodology and according empirical approaches. The existence of the given factor consequently implies an inherent multidisciplinary nature of the proactive computing paradigm. Consequently, after having identified the main characteristics of the empirical findings, we may assert that for the purposes of unlocking and using more effectively the potentials of the proactive computing, we have to include the expertise of the associated cognitive science domain in our methodology. Only then we are able to properly adapt the functional capacity of our system towards more intuitive and perceptive operative qualities.

Ultimately, the given empirical finding serves as the main basis and necessary guiding principle for the design and realisation of our second empirical study. During our successive experiment, we therefore aimed to demonstrate the *feasibility* of the concept realisation with respect to the implications of the previously identified results.

The aforementioned analytical outcomes of the first empirical study, consequently provide a stable methodological basis, which ultimately opens the doors for an implementation of more elaborate computational techniques and concept designs. The obtained results of the first experiment demonstrate that for the purposes of an effective concept evolution within the proactive computing domain, a system's methodological framework has to go beyond pure computer science principles and approaches. Consequently, the given factor implies a substantial theoretical shift of the applied methodology towards an interdisciplinary field, which initially includes the expertise of the cognitive science domain.

Implications of the second empirical study

The second empirical study allows us to establish a necessary proof of concept on a more intricate probabilistic level, which subsequently implies two following implications. (1) According to the functional specifications of the implemented probabilistic methods, such as the Bayesian parameter estimation, we may achieve a better accurateness of an event detection and overall context recognition [35]. (2) Subsequently, an intuitiveness or sensitivity level, related to the context characteristics may be achieved through a methodological reorientation towards an interdisciplinary domain. The given shift is attained through an application of the corresponding techniques, issued from the cognitive modelling approach.

For the purposes of a context monitoring, we used the deterministic mechanisms, which only allowed us to detect the predetermined instances of the related context details. The given mechanisms are based on a simple detection of the predefined context characteristics, through the corresponding proactive scenarios. The given approach characteristics consequently allowed us to identify and to observe only the superficial layers of an associated context. However, for a more intuitive proactive mediation, we need the more accurate tuning of our system towards hidden layers of a context or, in other words, towards underlying aspects of the user's cognitive states.

In the second empirical study, we therefore decided to design a more intuitive perceiving mechanism, allowing the proactive system to operate on a more subtle level of context details. On the one hand, through the running proactive scenarios and applied cognitive modelling methodology, we detect the predefined instances of a user's behaviour, which consequently reflect the variations of the user's currently prevailing cognitive characteristics. On the other hand, by means of the Bayesian formula, we estimate the quality and properties of the detected data, allowing to better define all corresponding specifications of the future proactive mediation, including time and type of actions [11]. The given approach allows us to significantly increase the quality and accurateness of both, the detected context data and assigned proactive actions.

Consequently, in order to increase an intuitiveness or sensitivity level of the system's monitoring mechanisms, we apply the additional empirical techniques, provided by the chosen methodology of the cognitive science domain. A higher intuitiveness level, in relation to the context events, is successively achieved by employing the associated techniques of the cognitive modelling approach [64]. The given approach provides us with the possibility to additionally take into consideration the underlying aspects of the user's cognitive characteristics.

In order to achieve the aforementioned goal, we model the user's behavioural patterns, which initially represent the variations of two cognitive states. The instances of a user's behaviour are subsequently detected and evaluated by a set of several corresponding proactive scenarios. In other words, the context monitoring process is divided in two main parts, deterministic and probabilistic. The deterministic part has the objective to detect the exact instances of a user's behaviour, whereas the probabilistic part aims to evaluate each instance of the identified data in order to estimate the properties of the currently prevailing context settings. The application of the given interdisciplinary approach consequently allows the system to penetrate into more subtle layers of context details, which at this point are represented by the hidden aspects of the user's cognitive variations.

The second empirical study ultimately allowed us to elaborate a more adapted methodology, which successively provided us with the additional techniques and methods of an interdisciplinary character. The chosen multi-domain approaches allowed us to build an early prototype version of the proactive system, characterised by a more accurate and intuitive functional capacity. Thus, the main results' implication of the second experiment highlights the feasibility of the concept realisation and consequently implies that the proactive computing may attain a higher degree of its potential, if applied within a multidisciplinary methodological framework.

5.2.3 Findings in relation to their general implications

From their general perspective, the experiments' results have several wider meanings and implications, which may be considered as the potential orientations for the existing and future proactive computing research. The given results' implications are briefly highlighted in the following paragraphs.

Benefits of the modular approach

Both applied design techniques of the deterministic and probabilistic approaches, in their general implementation specifications are stipulated by a unique methodological orientation. The given orientation is characterised by the proactive system's *modular approach*, which comprises several functional benefits. Moreover, the mentioned modular approach represents the core of the system's framework, which is primarily used as a functional basis for all additional methods and techniques. The application of the chosen design techniques consequently allows us (1) to separate the system's functions into semi-dependent processes, (2) to increase the productivity and functional capacity of the system without changing its core mechanisms, and (3) to extend the system's adaptability in relation to a variety of new target environments.

The obtained empirical findings indicate that the proactive system can be effectively based on the principles of a target context segmentation. In order to achieve an effective monitoring of a context environment, we must first provide a system with a basis of the predefined target elements or, in other words, instances of the modelled context events. In our opinion, the given approach is crucial, as it allows us to define the system's initial functional orientation. Additionally, an application of the mentioned strategies provides all further probabilistic techniques with a stable deterministic framework.

Consequently, in order to implement the aforesaid strategies of a target environment segmentation, we need a set of the corresponding proactive scenarios, which will cover the predefined part of context settings. The detected specified events can ultimately serve as a functional input data for all further probabilistic evaluations of a related context setting. In other words, we need the composite functions of the deterministic framework in order to provide our probabilistic model with the corresponding input values, reflecting the precise specifications of current context settings.

The given approach characteristics consequently reflect the main structure and underlying objectives of our two empirical studies. Notably, the first experiment aims to test our main monitoring mechanism, which is entirely built on the principles of the deterministic approach. On the other hand, after having formalised the functional operation of the deterministic framework, we were further able to implement our newly designed probabilistic mechanisms. Thus, during our second exploratory study, we aimed to test the combination of two frameworks. The first structure of the deterministic methods served as the main technique for providing the second structure with the required mechanisms of a data detection and parameters' supply.

Consequently, upon the probabilistic data evaluation, we are able to identify the concealed abstract details, and thus to specify the underlying properties of a target context environment. In other words, by applying the probabilistic mechanisms of a context data evaluation, we are able to refine the superficial characteristics of the associated context settings. Ultimately, as stipulated by the results of two experiments, the system's modular design and methodological structure of two frameworks allow the proactive system to considerably increase its accurateness level of context monitoring processes. Moreover, the combination of the aforementioned approaches provides the system with the according operational mechanisms, which allow the proactive module to increase the efficiency level of its functional adaptability.

The mentioned approach allows us to build a particularly adaptive system, which is capable to reorganise its functional orientation by means of the newly added proactive scenarios. In other words, more composite proactive scenarios the system contains, more context details it is capable to cover. The given approach implies that a functional operation of the proactive system can be easily adapted to any type of a context environment. Consequently, the potential context environments may comprise a variety of the target domains, including health monitoring systems, energy management systems, security and threat prevention systems, satellite operating and management systems, disaster prevention systems, air traffic and car traffic monitoring systems, financial and economic stability monitoring systems and so on. Therefore, in order to adapt the proactive system to a new target environment, we simply have to design and implement the according proactive scenarios, which will subsequently cover the various composite aspects of the new context environment. However, regardless of the number and variation of the implemented proactive scenarios, the Rules Running System and its operational principles remain unchanged.

The chosen approach of the modular design characterises the composite structure of the proactive system as highly generic. The key computational principles, represented by the Rules Running System remain uniform in all circumstances. Upon an extension of the system's capabilities, through new proactive modules, we never change or modify the main running mechanism of the proactive engine. Due to the system's modular structure, in order to extend its initial functional capabilities, it is simply enough to design the corresponding proactive scenarios, which will implement the new required features without modifying the system's main running mechanism. The given factor characterises the proactive system as a generic and adaptive mechanism, which consequently allows us, if needed, to reorganise its initial functional orientation by selecting the corresponding proactive scenarios. Alternatively, in order to extend the system's functional capabilities, we simply have to add the new supporting proactive scenarios.

Benefits of the interdisciplinary approach

Successively, the both empirical studies demonstrated a notable benefit and importance of applying the chosen interdisciplinary methodology. The first empirical study emphasised an important research factor, stipulating that for a more accurate and intuitive system's functional capacity, we need to include an expertise of the associated cognitive science domain in our methodological framework. On the other hand, the second empirical study described the precise conceptual characteristics and functional specifications of the applied interdisciplinary methodology, and thus demonstrated the feasibility of the concept realisation.

The chosen approach orientations, consequently provided the research project with the highly valuable methodological contributions. The empirical findings stipulate that in order to fully reveal the potentials of the proactive computing paradigm, we have to situate it within an interdisciplinary methodological structure. According to its initial definition, as proposed by David Tennenhouse, the proactive computing has to constantly deal with various context elements and their corresponding characteristics, including a multitude of human-related factors [1, 2].

As we saw in our study, the given human-related factors are essentially represented by a variation of the user's cognitive characteristics, which are largely manifested during the user's interactions with the associated elements of a context. In other words, better we want to comprehend the context composite aspects of a user's behaviour and its main influential factors, more cognitive-based approaches we need [53, 64, 82]. Therefore, as our study demonstrates, by employing the according multidisciplinary techniques, we are ultimately able to build an intuitive, user-perceptive, context-aware proactive system.

On the contrary, if we only apply the techniques of the computer science domain, we are solely able to cover the superficial layers of the associated context environment. On the other hand, by employing the additional cognitive modelling techniques, issued from the cognitive science domain, we are able to penetrate into more subtle abstract layers of the same context environment. In other words, the chosen cognitive modelling methodology allows us to identify and then to translate the hidden metaphysical aspects of an associated context into an algorithmic level of the system's composite functions. Consequently, the given multidisciplinary approach provides the system with a valuable mechanism, capable to identify and to account the hidden layers of context specifications.

5.2.4 Section summary

Our main objective in the given section is to analyse and to fuse the main results' implications from both empirical studies in a discursive form. We divided all analytical proceedings into three categories, with the underlying objective to analyse the experiments' results from three different perspectives. First, we reviewed the results' implications in relation the chosen methods and approaches of our methodological framework. Subsequently, we analysed the results in relation to our theoretical background and ultimately in relation to their general implications and meanings.

In the first subsection of the results' relation to the applied methods and approaches, we presented the main findings' implications, which accordingly addressed the formalisation aspects of the deterministic framework. The first implication of the empirical findings defined the currently employed proactive system concept as a functional and effective framework structure. The second implication stipulated that the implemented event-monitoring techniques of the proactive system required more intuitive mechanisms of an outward context analysis. The new sensory mechanisms have to be sensitive enough in order to be always able to detect the hidden metaphysical aspects of user's cognitive characteristics.

Subsequently, we have delineated the corresponding results' implications, which addressed the formalisation aspects of our probabilistic framework. The first implication of the empirical findings has stipulated that a more complex structure of the second-wave proactive scenarios may be effectively based on the principles of the probabilistic approach and cognitive modelling methodology. The second implication stipulated that the applied probabilistic and cognitive modelling techniques consequently necessitate a thorough parameter optimisation and overall models' refinement. In other words, we need to specify more accurately the composite function's parameters of the probabilistic model and to thoroughly adjust the modelled patterns of a user's behaviour.

Consequently, in our second subsection, we highlighted the main results' implications in relation to the established theoretical background. In the scope of the first experiment's results, we emphasised the following three derived implications. (1) According to results of the first experiment, the concept of the proactive computing, as it was defined by David Tennenhouse, has the positive effect on a quality and effectiveness of a user's overall performance. (2) According to the collected statistics and results of the system's general performance, we may state that it is technically possible to implement the concept of proactivity on the basis of the chosen methods and approaches. (3) Ultimately, in order to move beyond the predefined deterministic structure of the proactive system, we need to qualitatively extend its current conceptual and methodological framework by adapting new theories, methods and approaches.

Subsequently, in the scope of the second empirical study, we emphasised the following two derived implications. (1) According to the functional specifications of the implemented probabilistic methods, such as the Bayesian parameter estimation, we may achieve a better accurateness of a context event detection. (2) An intuitiveness or sensitivity level, related to context characteristics, may be achieved through a methodological re-orientation towards an interdisciplinary domain. The given shift is attained through an application of corresponding techniques, issued from the cognitive modelling approach.

Ultimately, in our last subsection, we highlighted the empirical findings in relation to their general implications and meanings. The first derived implication stipulated the functional benefit of applying the methodological aspect of the modular approach. The application of the chosen design techniques consequently allowed us (1) to separate the system's functions into semi-dependent processes, (2) to increase the productivity and functional capacity of the system without changing its core mechanisms, and (3) to extend the system's adaptability in relation to a variety of new target environments. Subsequently, the second results' implication stipulated a notable benefit of applying the chosen interdisciplinary methodology. The given approach orientations, consequently provided the research project with the highly valuable methodological contributions, which ultimately allowed us to build a more intuitive, user-perceptive, context-aware proactive system.

Chapter 6

Conclusions and future directions

The current chapter represents the conclusive part of our work, which aims to bring together all previously stated hypotheses, formulated research questions, applied methodological techniques and obtained empirical findings. The main objective of this chapter is to provide an ultimate highlight of all essential research fundamentals, applied principles and factual elements, allowing to provide the clear answers to the initially formulated research questions.

In order to clearly delineate the key points of the research, we proceed according to the initial organisation of our thesis. The given approach will allow us to provide a structural synopsis of all essential research factors in their respective presented order. Thus, in section 6.1 we start by highlighting the study's main points and key orientations of the research. Here, we restate our initial research questions. Additionally, in the same section we review the essential characteristics of proactive computing paradigm. In the subsequent section 6.2, we highlight the thesis' main methodological aspects together with the applied concept principles. After having reviewed the key specifications of our approach, we successively make a brief overview of our results' main implications in section 6.3. Consequently, with regard to main methodological aspects and obtained empirical findings, we highlight in section 6.4 several potential directions for the future research endeavours.

6.1 Study's main points and key research orientations

In order to review the initially formulated research directions, we provide in the given section a highlight of main study points, which successively defines the thesis' structure and its key orientations. We review our study area and we provide a short synopsis of its essential compound definitions. Subsequently, we restate our research questions in order to provide the conclusive remarks, regarding their underlying relations to the obtained empirical data.

6.1.1 Study area and research objectives

Throughout the current work, we have primarily investigated the domain of *Proactive computing*, which may be characterised as a comparatively new computer science research paradigm [2]. Conceptually, the architecture and main framework of our software system is entirely based on the concept characteristics of the mentioned approach. The notion of *proactivity* may be largely defined as an anticipation of a context event, activity-based problem, unwanted event, context conditions, or context change, which are detected for the purposes of performing proactively the necessary context mediating actions.

In our research, we understand and use the notion of *Proactive computing* as it was defined by Dr. David L. Tennenhouse. In his vision of future of the computer science research he explores a new paradigm, where he proposes to re-examine the relationships between physical and abstract domains of a context, to re-evaluate our general approach of interactive computing, and to gradually move from interactive human-centred to proactive human-supervised models of human-computer interactions [1, 2]. By taking into consideration the relationships between the associated aspects of physical and abstract domains of a context, and through the enhancement of a target system by a proactive type of behaviour we are therefore able to build more intuitive and perceptive software systems.

According to the definition of a proactive system, given by David Tennenhouse, we may highlight two main principles, which govern the entire process. The first principle stipulates that any proactive system is always working on behalf of a user, where the second principle specifies that a proactive system acts on its own initiative without explicit instructions from the user [1]. In other words, we can assume that the system may possess a set of policies, which define the patterns of its behaviour in various contextual situations. Therefore, as a general characteristic, in order to prevent an unwanted event from happening, a proactive system always aims to cause a change in a context, rather than just to react to changes [13, 14].

Consequently, the proactive computing as a new research paradigm doesn't possess yet a clear methodological support and approach repository, which may initially provide us with the necessary techniques, methods and tools for building a fully functional proactive system. Being a comparatively new research direction, an application of the proactive computing thus requires a thorough empirical and theoretical investigation. The undertaken research efforts allow us to reveal and to formalise the underlying functional characteristics of the approach together with its associated methodological traits. Therefore, the main objective of the present research project is to elaborate, build and test the valid methodological and empirical frameworks, which will subsequently allow us to provide our proactive system with a scientifically rationalised proof of concept.

Basing on the underlying factual traits of a proactive computing research, indicating the lack of the clear methodological support, we therefore undertake the construction of our own repository of guiding methods and approaches. Consequently, we decide to investigate the proactive computing paradigm within the defining principles of two computer science approach orientations. The chosen research directions respectively include deterministic and probabilistic methodological frameworks. In order to provide an according supportive basis for the system's further conceptual and technical developments, we define the corresponding functional principles for each applied approach framework.

On the deterministic level, we specify the according functional specifications of the proactive system, which stipulate that we have to anticipate a target context event, basing on the techniques of pre-programmed proactive scenarios and rules [10, 52]. The given approach has the initial objective to provide the stable system functioning with the first implemented prerequisites of the proactive computing paradigm. Correspondingly, on the probabilistic level, we specify that the system has to anticipate a new event or to refine the context details of an initial event, basing on a probabilistic analysis of already available context information [11]. In parallel to the mentioned event monitoring processes, the system has to additionally identify an exact activation time for the potential response-actions, related to the detected context event.

According to the aforementioned methodological specifications, the implemented mechanisms of the deterministic approach provide the necessary functional grounds for the corresponding mechanisms of the probabilistic approach. In other words, the stochastic mechanisms of our probabilistic framework are conceptually linked to, and based on the functional output of the elaborated deterministic framework. In order to refine the prevailing context details and thus to penetrate into more subtle abstract layers of context characteristics, we need first to identify the initial prerequisites of a target event. The identified and collected context data must be the exact representation of the initial target event. The given approach allows us to provide the corresponding system's algorithms with the event's original exact properties for all further probabilistic evaluations of context abstract details.

In order to detect the predetermined target events, the context monitoring mechanisms of the proactive system have to be based on the underlying principles of the deterministic approach. Furthermore, in order to be able to penetrate into more subtle and deeper layers of prevailing context details, the system's analytical mechanisms must be accordingly based on the underlying principles of the probabilistic approach. Our main objective for applying the stochastic mechanisms of a data evaluation is to be able to precise the quality and properties of currently prevailing context characteristics, which are initially represented by various detected instances of the predefined target events.

Consequently, in order to fulfil the thesis' initial objective, we have to validate, or in other words to formalise the theoretical and empirical specifications of both methodological frameworks. The given approach will subsequently allow us to provide the necessary proof of concept for all applied design aspects and their corresponding technical realisations. Additionally, we are able to formalise the proactive system's general structure, its underlying methodological framework, and eventually the quality of its primary functional capacities. Due to the thesis' main objective, stipulating the necessity to scientifically rationalise all proactive system's operating mechanisms, we have to design, formalise, and test both, the governing theoretical and empirical aspects of our entire methodological framework. The given objective leads us towards the specification of the thesis' main research problem and all associated research questions, which collectively define the underlying structure of the current work.

6.1.2 The scope and specifications of the research problem

Research problem. As we build our system, conceptually basing on a new research dimension of proactive computing paradigm, our proactive system does not possess a clear empirical and methodological support. Throughout our literature research, we have identified the lack of coherent specifications of the domain approach repository and essential methodological definitions. Therefore, for the purposes of an effective formalisation of the system's applied methods and approaches, we need a more extensive investigation on theoretical and empirical levels. The given objective allows us to provide a valid proof of concept for all designed conceptual elements and implemented functional characteristics of our proactive system.

Consequently, in order to address the aforementioned research problem, and thus to provide a valid proof of concept for our proactive system, we subsequently formulate a set of research questions, which aim to guide us throughout the progress of our research. Notably, the stated research questions define the main directions of our study. Each formulated research question aims to address a specific aspect of the identified research problem. Collectively, all research questions define three main orientations of the applied methodological framework, including the theoretical basis, approach directions on a deterministic level and approach directions on the corresponding probabilistic level.

Research question #1. Our first research question aims to determine the theoretical basis and methodological prerequisites for all further specifications of the applied methods and approaches. Thus, the Research Question #1 is formulated as follows:

> *What are the main theoretical and methodological prerequisites and orientations, which define the conceptual directions of a system, built within the framework of the proactive computing paradigm?*

Consequently, we provide the corresponding answer, which specifies that on the theoretical level the proactive system's conceptual aspects are mainly stipulated by the composite characteristics of several related computer science approaches. Besides the proactive computing itself, the list includes the theoretical orientations of ubiquitous computing, autonomic computing and context-aware computing [2, 16, 32]. Subsequently, on the methodological level the system's conceptual orientation is stipulated by two approaches of the corresponding deterministic and probabilistic research frameworks. The chosen methodological disposition is partially conditioned by an analysis of the related research literature.

Research question #2. Our second research question aims to specify the main conceptual characteristics and design orientations, which subsequently govern the proactive system's development process within the deterministic paradigm. The Research Question #2 is formulated as follows:

> *What are the key principles and elements, which define the main system concept, its functionality and its integrity on the fundamental deterministic level?*

Correspondingly, we provide an answer to our second research question, which stipulates that on the deterministic level the system's main design orientations are essentially characterised by various functional attributes. The list comprises several fundamental elements of the proactive system, including Rules Running System, proactive scenarios and rule's algorithmic architecture [5, 12]. Consequently, the validation or formalisation of the deterministic side of the system is primarily achieved through our first empirical study, which assesses the relevance, quality, reliability and the outward implications of the aforementioned functional attributes.

Research question #3. Our last research question aims to determine the main conceptual characteristics and design orientations, which accordingly specify the proactive system's transition to the probabilistic dimension. The Research Question #3 is formulated as follows:

> *What are the key principles and approaches, which define the transition of the framework concept from deterministic towards probabilistic dimension?*

Consequently, we provide an answer to our third research question, which stipulates that on the successive probabilistic level the system's main design orientations are characterised by several implemented approaches. The corresponding list of underlying empirical methods includes several applied techniques, including *Cognitive modelling approach* and *Bayesian parameter estimation* [11, 58]. The validation or formalisation of the probabilistic side of the system is achieved through our second empirical study, which explores various interdisciplinary techniques.

6.2 Synopsis of key methodological aspects

After having reviewed the thesis' main orientations and objectives, we present below the successive procedural steps, which collectively describe all theoretical and empirical aspects of the thesis. In the given section we highlight the project's key methodological factors, which constitute the main body of the thesis. Our methodological framework is the structural representation of all composite theoretical, conceptual and empirical aspects, necessary for a realisation and accomplishment of the current work.

6.2.1 Characteristics of the concept formalisation

The key methodological specifications of the present study are characterised by a formalisation of the chosen concepts and methods within the principles of two theoretical perspectives. The application of the given approach is fundamental as it allows us to build the proof of concept in a progressive manner. We conceptualise, elaborate and implement our methods first by following the principles of the deterministic approach. The given methodological aspect provide us with the possibility to build the stable fundamental basis of the proactive system. Consequently, the formalised functional basis allows us to implement more evolved, context monitoring and data analytical techniques. Thus, we elaborate and implement the probabilistic part of our system, which is conceptually linked to the underlying methods of the deterministic framework. In other words, by applying the probabilistic computational techniques, we try to extend the system's initial context monitoring capabilities and thus to increase the accurateness and intuitiveness level of a successive proactive mediation.

The implementation of the fundamental deterministic framework allows us first to investigate the functionality and potentials of the key principles of basic proactive functions and their successive outward impact. On the other hand, the implementation of the second probabilistic framework allows us to go further, where we able to explore the fundamental principles of the proactive computing paradigm on more advanced levels.

The realisation of the deterministic framework is essentially based on the design and application of several computational techniques, which provide the fundamental system's functioning. In order to implement the underlying proactive computing principles, we elaborate and apply a number of key deterministic elements, including Rules Running System, proactive rule and its fixed algorithmic structure, proactive scenarios and their inherent task delegation mechanisms [5, 10]. For the purposes of a context monitoring function, we design several proactive scenarios, which are programmed to capture the predefined instances of corresponding context conditions. Conceptually, a proactive scenario is represented as a set of compound predefined rules, which specify the scenario's main type and its basic functional characteristics. The collections of predefined rules, represented by specific proactive scenarios are consequently executed by our Rules Running System, which ensures the continuity aspect of the proactive system's monitoring and mediating processes. Therefore, all aforementioned composite elements, built within the principles of the deterministic framework, collectively provide a stable operational basis for all further developments of the system's functional concepts.

An implementation of the stable deterministic framework allows us to design and to implement more elaborate context monitoring and context mediating techniques, based on the principles of a probabilistic approach. Conventionally, the probabilistic framework consists of several composite proactive modules, which collectively allow us to refine the accurateness of the system's monitoring capabilities and to increase the intuitiveness level of the system's mediating actions. The aforementioned framework includes an implementation of the probabilistic data evaluation by means of the applied Bayesian parameter estimation. Additionally, the framework includes an adaptation of related interdisciplinary techniques, borrowed from the cognitive modelling approach [53, 58]. The application of the mentioned approach consequently allows us to create an interdependent link between abstract and physical domains of a target context environment.

The applied techniques of a multidisciplinary methodology provide us with the possibility to model the user's essential behavioural characteristics, allowing the system to detect a variation of users' prevailing cognitive states [57, 64]. Upon the realisation of the given approach, we are able to translate the manifested instances of the user's mental characteristics into an algorithmic level of the system's functions. Consequently, the Bayesian parameter estimation is applied upon the detected context data, allowing to evaluate the prevalence level of the user's manifested cognitive states. Therefore, the application of both approaches provides us with the possibility to penetrate into more subtle abstract layers of a context, and thus to increase the accurateness and intuitiveness level of the system's monitoring and mediating techniques.

A formalisation of both methodological approaches plays the crucial role in an overall objective of providing our system with a valid proof of concept. An effective validation of the chosen concepts, theories and methods, allows us to provide an empirical evidence, necessary for a well grounded proof of concept. Conventionally, a methodological structure of the concept validation is represented as a gradual formalisation process of all system's aspects. First, we elaborate, design and formalise the system's fundamental mechanisms through an implementation of essential computational techniques, basing on the underlying principles of the deterministic approach. Second, we elaborate and formalise the successive stochastic mechanisms of a context data evaluation through an implementation of associated computational techniques of the probabilistic approach. Ultimately, the effective validation of both methodological frameworks and their corresponding computational techniques, provides our system with the necessary, scientifically rationalised proof of concept.

6.2.2 Specifications of the concept implementation

A formalisation of the system's general concept consequently requires the elaboration of corresponding applicable methods and approaches, allowing to incorporate the underlying functional principles of the prospective methodological frameworks. As we have seen in our earlier chapters, one of the fundamental elements of the proactive system is a proactive scenario. The underlying mechanisms of the implemented proactive scenarios ultimately provide our system with the valuable proactive behaviour. A variation of the designed and implemented proactive scenarios defines an orientation of the system's functional characteristics. Therefore, in order to build our proactive system, which is conceptually based on two distinct methodological frameworks, we have to design two distinct variations of the system's proactive scenarios.

The formalisation of both methodological frameworks is obtained through the design, implementation and testing of two waves of proactive scenarios. The main design orientations and conceptual characteristics of the first-wave proactive scenarios aim to define the fundamental algorithmic mechanisms, which allow us to implement the basic functional capacity of a system's proactive behaviour. Moreover, by formalising the underlying mechanisms of the first-wave scenarios, we are able to create a stable functional framework for all future types of proactive scenarios. Thus, the first-wave proactive scenarios are mainly characterised as a fundamental realisation of the initial concept, which is entirely based on the underlying principles of the deterministic approach.

Correspondingly, the main design objectives of the second-wave proactive scenarios aim to elaborate more accurate operative techniques, allowing to increase the intuitiveness and efficacy level of the system's monitoring and mediating capabilities. The given set of proactive scenarios is based on, and designed according to the functional principles of the first-wave scenarios. However, the implementation of additional probabilistic and cognitive modelling techniques distinguishes both types of proactive scenarios in terms of their complexity, functional capacity and the acquired qualities of the system's monitoring and mediating capabilities.

6.3 An overview of experiments' objectives and results' main implications

As we saw in our theoretical and methodological descriptions, a formalisation process of the proactive system's key mechanisms is consequently achieved by means of the organised empirical studies, which aim to investigate various aspects of the implemented methods, techniques and approaches. Both waves of proactive scenarios, together with their underlying functional mechanisms are therefore subjected to a thorough investigation, performed within the framework of two experiments.

The key objectives of the first experiment consisted of organising an *enquiry study*, which aimed to test the system's fundamental functions and their overall performance, to evaluate the system's functional capacity and to study an impact of a system's proactive behaviour on users' general performance. In other words, the first empirical study aimed to validate the deterministic part of the proactive system and, on the other hand, to determine the scale of its functional impact on a user. Ultimately, an effective formalisation of the deterministic mechanisms opened the doors for a successive implementation of various supplemental techniques of the probabilistic approach.

Consequently, the key objective of the second experiment consisted of organising an *exploratory study*, which aimed to elaborate the system's initial functional capabilities by applying more advanced techniques of the probabilistic approach and cognitive modelling methodology [11]. In other words, the second empirical study aimed to validate the probabilistic part of the proactive system and, on the other hand, to emphasise the feasibility of a concept realisation and to highlight a value of the multidisciplinary methodological contribution. Ultimately, from a theoretical point of view, the second experiment aimed to outline the potential directions for the future research efforts.

Throughout the length of an entire research process, the current study aims to emphasise the importance of situating the proactive computing paradigm within an interdisciplinary methodological framework. In our opinion, the given approach allows us to significantly increase the quality and scale of proactive computing potentials. Besides a formalisation of two methodological frameworks, both enquiry and exploratory empirical studies aim to respectively highlight first the importance and second the feasibility of a multidisciplinary methodological structure.

Upon the first *enquiry study*, we have identified the following empirical findings.

(1) Proactive computing is possible, if applied within the defined fundamental deterministic structure [10].

(2) Proactive system behaviour positively affects the users' general performance [52].

(3) User-oriented proactive scenarios prove to be an efficient system's mechanism for mediating a target context environment.

(4) For a more effective performance, the proactive system's context monitoring and context mediating mechanisms require more accuracy and intuitiveness in relation to the hidden aspects of users' cognitive states.

Given the identified empirical findings, we may consequently notice a presence of the proactive system's strong methodological inclinations towards a systematic consideration of a cognitive science expertise. The designated methodological orientation towards a more extensive application of various cognitive-based approaches has been essentially stipulated by insufficient perceptive qualities of a basic proactive system's behaviour in relation to the hidden aspects of users' cognitive states.

Consequently, during our next empirical study, we aimed to take further the outcomes of the first experiment and thus to enhance the system's perceptive capabilities by introducing new techniques of the cognitive modelling approach and probabilistic data evaluation. Ultimately, upon our successive *exploratory study*, we have identified the following empirical findings.

(1) A more intuitive proactive system behaviour is possible, if realised within the defined interdisciplinary methodological framework.

(2) It is technically feasible to increase the system's potential and general proactive computing proficiency, if applied in conjunction with the cognitive modelling approach and probabilistic data evaluation [11].

(3) The given probabilistic framework requires a more accurate and meticulous optimisation of its parameters as it touches more complex factors of users' cognitive processes and deals with the hidden metaphysical aspects of context settings.

Given the identified empirical findings of our second exploratory study, we may notice that an application of the cognitive science expertise plays the crucial role in an overall process of refining the initial qualities of a proactive system's behaviour. Upon the given phase of a concept refinement, the cognitive science expertise takes a leading role in a quality optimisation process, which consequently allows us to partially unlock the hidden values of proactive computing potentials. The designated methodological orientations are essentially stipulated by an increasing need to include a cognitive science expertise, which ultimately allows us to adequately deal with the underlying human-related factors of a target context environment.

After having reviewed all identified empirical findings, we may subsequently emphasise an inherent interdisciplinary nature of the proactive computing paradigm. An effective implementation of the elaborated interdisciplinary methodological framework provides a valid, scientifically rationalised proof of concept and a stable ground for all future similar research orientations within the proactive computing domain.

6.4 Future research directions

According to the earlier discussed methodological and empirical characteristics of our study, we may at present assert that at the given state of the research we have the fully functional proactive system, which is based on the fundamental principles of the deterministic approach. Furthermore, we have the first, early prototype of the proactive system, which is based on the underlying principles of the probabilistic approach. The first empirical study allowed us to validate the fundamental mechanisms of the proactive system's deterministic framework. Concurrently, we were able to optimise the system's composite functions and to assess the quality of their impact on users' general performance, which consequently allowed us to produce the fully functional prototype of the proactive system. Accordingly, the second exploratory study allowed us to formalise and to implement the composite elements of the system's successive probabilistic framework. However, due to the imposed time limitations we were not able to fully optimise the system's probabilistic mechanisms and to empirically assess the quality of their functional impact on a user. Therefore, we have to attribute the system's optimisation process and functional assessment to the future research efforts.

The main orientations for the future research are primarily characterised by an underlying necessity of organising the additional enquiry study, which will allow us to attune the system's functional mechanisms and to empirically assess the quality of their outward impact. The implemented mechanisms of our probabilistic framework will successively require a more accurate functional optimisation of the defined models' parameters and patterns' specifications.

The first key objective of the potential third experiment will be an optimisation process of the system's functional properties, which specify the composite aspects of both cognitive models. Due to general characteristics of the present study, the currently implemented simulations of users' behavioural patterns are modelled on a basis of theoretical postulations of the cognitive psychology and not on a basis of the real empirical data. However, the elaborated cognitive models represent the most probable variations of a user's browsing behaviour at the different context conditions. In order to empirically support the modelled user's behavioural patterns, we need to attest our cognitive models by the real, user-based empirical data, which will define the main characteristics of a user's browsing activity at different context conditions.

In order to achieve the stated objective we will have to collect the real user's browsing data, which will subsequently help us to clearly define the composite specifications for both cognitive models. We will have to record and to analyse all users' manifested behavioural patterns, which initially express the users' cognitive states of mental satisfaction or dissatisfaction. All collected instances of a user's behaviour will have to be generalised and sorted according to the quality of their expressive characteristics. Given the underlying specifications of the aforementioned experiment, we will need a more extensive theoretical and empirical input from the related methodologies of the cognitive psychology.

The described characteristics of the first objective ultimately lead us to the successive second objective of the potential future experiment. Upon the identification and collection of an empirical user's data, we project to additionally assess the already implemented mechanisms of the current probabilistic framework. We will have to test the performance, quality and an outward impact of all newly implemented system's functions, dealing with the detection and analysis of the cognitive models. The empirical investigations will be dedicated to a quality assessment of the proactive system's functional impact on a user's browsing activity and search performance. In order to effectively fulfil the aforementioned objective, the empirical study will have to include the prerequisite control and study group participants.

The future research may concurrently include various prospective directions for a progressive extension of the proactive system's current functional capabilities and built-in features. We intend to dedicate our future research efforts to the realisation of several specific objectives, including:

- Elaborating the necessary mechanisms, which will allow us to implement the feature of an event priority for the dedicated proactive actions.
- Elaborating a mechanism, which will consequently allow the proactive scenarios to communicate between each other and, if necessary, to exchange the related context data.
- Designing and implementing a supportive, decision-making module, which will supervise a progression status of all running proactive scenarios and, if necessary, will be capable to intervene in case of context conflicts or action priority conflicts.
- Designing more advanced autonomic computing features, allowing the system to choose and activate the necessary proactive scenarios in order to adequately respond to new context conditions.
- Elaborating more extensive mechanisms, covering the proactive system's techniques of a user's data analysis, event prediction and general context evaluation.
- Extending the current methodological framework by incorporating the additional methods and approaches, borrowed from a theoretical repository of the cognitive psychology.

The delineated list of possible study directions is not exhaustive and thus does not represent the final ideas and orientations for our future research. The given list of prospective research efforts only highlights the main directions, which may subsequently be modified or extended.

6.5 Conclusion

The earlier described conceptual, theoretical and empirical characteristics of the current research project, collectively represent the thesis' methodological cornerstone. The elaborated methodological mechanisms aimed to provide tangible answers to all research questions and thus to generate the valid proof of concept. The formalised methodological structure consequently allowed us to scientifically rationalise the implemented functional principles of the proactive system.

The realisation of the aforementioned objectives may naturally take different conceptual forms and different research directions. In the case of the present work, the chosen methodological framework was distinctly characterised by an interdisciplinary aspect of the applied research methods. We supported the employed principles of the proactive computing paradigm by various interdisciplinary techniques. The tools and expertise of one research domain ultimately completed the methodological framework of the second research domain. The chosen approach orientations define the current study as a multidisciplinary research project, which in our opinion, is a highly important investigative and exploratory factor in the contemporary computer science research.

Today, we may easily notice a progressive expansion of various computer technologies, which successively pervade the daily routines of our lives. The contemporary computer science research comes across a requisite condition, which prompts us to start taking into account the hidden aspects of omnipresent human factors. The given evolutionary traits consequently imply the progressive need to include into computer science research the additional theoretical input and supporting expertise of the corresponding cognitive science research. The designated approach orientations will subsequently allow us to produce viable computer technologies, which will take into account the variations of a human situated behaviour and pervading subtle aspects of human cognitive states. Only thus we are able to produce truly intuitive, user-perceptive computer technologies.

The present study had the objective to emphasise the given link between computer science and cognitive science domains. Throughout the design and formalisation of our interdisciplinary methodological framework, we applied the domain-specific expertise from the related methodological and theoretical repository of the cognitive science research. We have emphasised in our study that a proactive computing paradigm is inherently linked to the hidden human factors, which compose the associated settings of a context environment. Consequently, during our experiments we have identified an important empirical finding, which stipulates that the proactive computing paradigm may display its hidden functional potentials, if applied within an interdisciplinary methodological framework. In other words, if we support the proactive computing paradigm, by a theoretical and methodological expertise of the cognitive psychology, we may eventually expand the boundaries of its initial functional capacities.

In order to provide a viable empirical basis, we had to build a scientifically rationalised proof of concept for our proactive system, which would substantiate the chosen conceptual orientations. All designed and implemented methodological mechanisms, together with the realised empirical studies, aimed to formalise the underlying concept characteristics of our proactive system. Consequently, an effective realisation of the system's proof of concept, together with the obtained empirical results, allowed us to assert that proactive computing is indeed tightly linked to underlying human factors. In order to display its full potential, the proactive computing research needs to be supported by the corresponding methodological and theoretical expertise of the cognitive science domain.

Besides the formalisation of the twofold methodological structure of the proactive system, the present research project allowed us furthermore to provide a viable basis for the future research endeavours. The achievements of the current work provide the future research and all related exploratory studies with a stable and operational methodological framework, which aims to describe the mechanisms of unlocking the proactive computing potentials. By providing a stable methodological and empirical basis, we are therefore able to support the prospective research efforts, which aim to explore the potentials of the proactive computing paradigm.

Bibliography

[1] David L. Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5): 43–50, May 2000. ISSN 0001-0782.

[2] R. Want, T. Pering, and D. Tennenhouse. Comparing autonomic and proactive computing. *IBM Systems Journal*, 42(1):129–135, 2003. doi: 10.1147/sj.421.0129.

[3] Sergio Coronado and Denis Zampunieris. Continuous Proactivity in Learning Management Systems. In *IEEE EDUCON 2010 Conference*, pages 199–204, 2010. ISBN 978-1-4244-6568-2.

[4] Denis Zampunieris. Implementation of a Proactive Learning Management System. In *E-Learn World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education*, pages 3145–3151, Hawaii, 2006.

[5] Denis Zampunieris. Implementation of Efficient Proactive Computing Using Lazy Evaluation in a Learning Management System. *International Journal of Web-Based Learning and Teaching Technologies*, 3(1):103–109, 2008.

[6] Jing Zhang, Jinhwan Lee, and Kwei-Jay Lin. Context-aware proactive process reconfiguration in service-oriented architecture. *Commerce and Enterprise Computing (CEC), 2012 IEEE 14th International Conference on*, pages 62–69, 2012. doi: 10.1109/CEC.2012.19.

[7] Yagil Engel and Opher Etzion. Towards proactive event-driven computing. In *Proceedings of the 5th ACM international conference on Distributed event-based system*, DEBS '11, pages 125–136, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0423-8. doi: 10.1145/2002259.2002279. URL http://doi.acm.org/10.1145/2002259.2002279.

[8] R. Helaoui. Towards a proactive system based on activity recognition. *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 849–850, 2010. doi: 10.1109/PERCOMW.2010.5470560.

[9] Babis Magoutas, Gregoris Mentzas, and Dimitris Apostolou. Proactive situation management in the future internet: The case of the smart power grid. volume 0, pages 267–271, 08 2011. ISBN 1529-4188. URL http://doi.ieeecomputersociety.org/10.1109/DEXA.2011.67.

[10] Sandro Reis, Denis Shirnin, and Denis Zampunieris. Design of proactive scenarios and rules for enhanced e-learning. In *CSEDU 2012 - Proceedings of the 4th International Conference on Computer Supported Education*, pages 253–258, Porto, Portugal, 2012. SciTePress.

[11] Denis Shirnin and Denis Zampunieris. Evaluating the probabilistic side of proactive computing using cognitive modelling methodology and the statistical inference of user's cognitive states. In *13th IEEE International Conference on Cognitive Informatics & Cognitive Computing*, London, UK, Forthcoming 2014.

[12] Marc El Alami, Nicolas Casel, and Denis Zampunieris. An architecture for elearning system with computational intelligence. *The Electronic Library*, 26(3):318–328, 2008. doi: 10.1108/02640470810879473. URL http://hdl.handle.net/10993/17179.

[13] George Vlahakis and Dimitris Apostolou. Proactivity in service based applications. *Informatics (PCI), 2012 16th Panhellenic Conference on*, pages 62–67, 2012. doi: 10.1109/PCi.2012.59.

[14] Mario G. C. A. Cimino, Beatrice Lazzerini, Francesco Marcelloni, and Alessandro Ciaramella. An adaptive rule-based approach for managing situation-awareness. *Expert Systems with Applications*, 39(12):10796–10811, 9 2012. doi: http://dx.doi.org/10.1016/j.eswa.2012.03.014. URL http://www.sciencedirect.com/science/article/pii/S0957417412004848.

[15] Jongyi Hong, Eui-Ho Suh, Junyoung Kim, and SuYeon Kim. Context-aware system for proactive personalized service based on context history. *Expert Systems with Applications*, 36(4):7448–7457, 5 2009. doi: http://dx.doi.org/10.1016/j.eswa.2008.09.002. URL http://www.sciencedirect.com/science/article/pii/S0957417408006301.

[16] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

[17] Daniel M. Russell, Norbert Streitz, and Terry Winograd. Building disappearing computers. *Communications of the ACM*, 48(3):42–48, 2005.

[18] D. Lupiana, C. O'Driscoll, and F. Mtenzi. Taxonomy for ubiquitous computing environments. *Networked Digital Technologies, 2009. NDT '09. First International Conference on*, pages 469–475, 2009. doi: 10.1109/NDT.2009.5272068.

[19] Ohbyung Kwon, Keedong Yoo, and Euiho Suh. Ubidss: a proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies. *Expert Systems with Applications*, 28(1):149–161, 1 2005. doi: http://dx.doi.org/10.1016/j.eswa.2004.08.007. URL http://www.sciencedirect.com/science/article/pii/S0957417404000983.

[20] M. Satyanarayanan. Pervasive computing: vision and challenges. *Personal Communications, IEEE*, 8(4):10–17, Aug 2001. doi: 10.1109/98.943998.

[21] S. Vansyckel, D. Schafer, G. Schiele, and C. Becker. Configuration management for proactive adaptation in pervasive environments. *Self-Adaptive and Self-Organizing Systems (SASO), 2013 IEEE 7th International Conference on*, pages 131–140, 2013. doi: 10.1109/SASO.2013.28.

[22] P. Horn. Autonomic computing: Ibm's perspective on the state of information technology. IBM Corporation, October 2001.

[23] K. Ahuja and H. Dangey. Autonomic computing: An emerging perspective and issues. *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, pages 471–475, 2014. doi: 10.1109/ICICICT.2014.6781328.

[24] Zhenxing Zhao, Congying Gao, and Fu Duan. A survey on autonomic computing research. *Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on*, 2:288–291, 2009. doi: 10.1109/PACIIA.2009.5406605.

[25] P. Lin, A. MacArthur, and J. Leaney. Defining autonomic computing: a software engineering perspective. *Software Engineering Conference, 2005. Proceedings. 2005 Australian*, pages 88–97, 2005. doi: 10.1109/ASWEC.2005.19.

[26] G. Martinovic and B. Zoric. E-health framework based on autonomic cloud computing. *Cloud and Green Computing (CGC), 2012 Second International Conference on*, pages 214–218, 2012. doi: 10.1109/CGC.2012.36.

[27] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 01 2007. URL http://inderscience.metapress.com/content/1184787H28163T15.

[28] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992. URL http://doi.acm.org/10.1145/128756.128759.

[29] B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22–32, 1994. doi: 10.1109/65.313011.

[30] N. Ryan, J. Pascoe, and D. Morse. Enhanced reality fieldwork: the context aware archaeological assistant. In *Proceedings of the 25th Anniversary Conference on Computer Applications and Quantitative Methods in Archaeology*, pages 269–274, University of Birmingham, 1997. Archaeopress.

[31] P. J. Brown. The stick-e document: A framework for creating context-aware applications. In *Proceedings of EP'96, Palo Alto*, pages 259–272, 1996.

[32] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *Proceedings of the CHI 2000 Workshop on "The What, Who, Where, When, Why and How of Context-Awareness"*. Georgia Institute of Technology, 2000.

[33] Negar Hariri, Bamshad Mobasher, and Robin Burke. Query-driven context aware recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 9–16, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2409-0. doi: 10.1145/2507157.2507187. URL http://doi.acm.org/10.1145/2507157.2507187.

[34] Jae-Kwon Kim, Kyoung-Hak Lee, Dong-Kyun Park, and Eun-Young Jung. Context-aware u-health service: Identification of exercise recommendation factors and creation of decision-making model using association rule. volume 0, pages 1–4, 06 2013. ISBN 978-1-4799-0602-4. URL http://doi.ieeecomputersociety.org/10.1109/ICISA.2013.6579439.

[35] Padipat Sitkrongwong, Saranya Maneeroj, and Atsuhiro Takasu. Latent probabilistic model for context-aware recommendations. volume 1, pages 95–100, 11 2013. URL http://doi.ieeecomputersociety.org/10.1109/WI-IAT.2013.14.

[36] G. S. Thyagaraju, M. M. Math, U. P. Kulkarni, and A. R. Yardi. Conflict resolving algorithms to resolve conflict in multi-user context-aware environments. *Advance Computing Conference, 2009. IACC 2009. IEEE International*, pages 202–208, 2009. doi: 10.1109/IADCC.2009.4809007.

[37] Jose Bringel Filho and Nazim Agoulmine. A Quality-Aware Approach for Resolving Context Conflicts in Context-Aware Systems. *2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing*, pages 229–236, October 2011.

[38] M. H. Tran, A. Colman, Jun Han, and Hongyu Zhang. Modeling and verification of context-aware systems. *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific*, 1:79–84, 2012. doi: 10.1109/APSEC.2012.50.

[39] M. Hussein, J. Han, A. Colman, and Jian Yu. An architecture-based approach to developing context-aware adaptive systems. *Engineering of Computer Based Systems (ECBS), 2012 IEEE 19th International Conference and Workshops on*, pages 154–163, 2012. doi: 10.1109/ECBS.2012.13.

[40] Ohbyung Kwon. The potential roles of context-aware computing technology in optimization-based intelligent decision-making. *Expert Systems with Applications*, 31(3):629–642, 10 2006. doi: http://dx.doi.org/10.1016/j.eswa.2005.09.075. URL http://www.sciencedirect.com/science/article/pii/S0957417405002757.

[41] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval. Context-aware recommender systems for learning: A survey and future challenges. *Learning Technologies, IEEE Transactions on*, 5(4):318–335, 2012. doi: 10.1109/TLT.2012.11.

[42] Zhang Yujie and Wang Licai. Some challenges for context-aware recommender systems. *Computer Science and Education (ICCSE), 2010 5th International Conference on*, pages 362–365, 2010. doi: 10.1109/ICCSE.2010.5593612.

[43] Yagil Engel, Opher Etzion, and Zohar Feldman. A basic model for proactive event-driven computing. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 107–118, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1315-5. doi: 10.1145/2335484.2335496. URL http://doi.acm.org/10.1145/2335484.2335496.

[44] P. Bedi and S. K. Agarwal. A situation-aware proactive recommender system. *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, pages 85–89, Dec. 2012. doi: 10.1109/HIS.2012.6421314.

[45] S. Agarwal, A. Singhal, P. Bedi, E. Jain, and G. Gupta. Proactive predictions to handle issues in recommendations. *Advance Computing Conference (IACC), 2014 IEEE International*, pages 555–560, Feb. 2014. doi: 10.1109/IAdCC.2014.6779385.

[46] D. Gallego, E. Barra, S. Aguirre, and G. Huecas. A model for generating proactive context-aware recommendations in e-learning systems. *Frontiers in Education Conference (FIE), 2012*, pages 1–6, Oct. 2012. doi: 10.1109/FIE.2012.6462246.

[47] D. Gallego, E. Barra, A. Gordillo, and G. Huecas. Enhanced recommendations for e-learning authoring tools based on a proactive context-aware recommender.

Frontiers in Education Conference, 2013 IEEE, pages 1393–1395, Oct. 2013. doi: 10.1109/FIE.2013.6685060.

[48] Toomas Timpka. Proactive health computing. *Artificial Intelligence in Medicine*, 23(1):13–24, 8 2001. doi: http://dx.doi.org/10.1016/S0933-3657(01)00073-2. URL http://www.sciencedirect.com/science/article/pii/S0933365701000732.

[49] G. Fenza, D. Furno, and V. Loia. Enhanced healthcare environment by means of proactive context aware service discovery. volume 0, pages 625–632, 03 2011. ISBN 1550-445X. URL http://doi.ieeecomputersociety.org/10.1109/AINA.2011.100.

[50] Xu Chao and Feng Zhiyong. A trusted affective model approach to proactive health monitoring system. *Future BioMedical Information Engineering, 2008. FBIE '08. International Seminar on*, pages 429–432, Dec. 2008. doi: 10.1109/FBIE.2008.52.

[51] Leilei Zhu, Zhendong Qian, and Jianhua Guo. Framework for developing proactive traffic control systems. *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 8:601–605, July 2010. doi: 10.1109/ICCSIT.2010.5565066.

[52] Denis Shirnin, Sandro Reis, and Denis Zampunieris. Experimentation of Proactive Computing in Context Aware Systems: Case Study of Human-Computer Interactions in e-Learning Environment. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2013 IEEE International Multi-Disciplinary Conference on*, pages 269–276, San Diego, US, February 2013. doi: 10.1109/CogSIMA.2013.6523857.

[53] Bruno Emond and Robert L West. Cyberpsychology: a human-interaction perspective based on cognitive modeling. *Cyberpsychology & Behavior: The Impact Of The Internet, Multimedia And Virtual Reality On Behavior And Society*, 6(5):527–536, 2003.

[54] B. R. Newell and A. Bröder. Cognitive processes, models and metaphors in decision research. *Judgment and Decision Making*, 3(3):195–204, 2008. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-70349176874&partnerID=40&md5=2ce64ad3260619e7acb32a937ae49396.

[55] Michael W. Eysenck and Mark T. Keane. *Cognitive Psychology: A Student's Handbook*. Psychology Press, Hove, 2000.

[56] Ron Sun and Nicholas Wilson. A model of personality should be a cognitive architecture itself. *Cognitive Systems Research*, 29–30(0):1–30, 9 2014. URL http://www.sciencedirect.com/science/article/pii/S1389041714000114.

[57] Gerhard Strube. Generative theories in cognitive psychology. *Theory & Psychology*, 10(1):117–125, 2000. URL http://tap.sagepub.com/content/10/1/117.abstract.

[58] Ron Sun. Theoretical status of computational cognitive modeling. *Cognitive Systems Research*, 10(2):124–140, 6 2009. doi: http://dx.doi.org/10.1016/j.cogsys.2008.07.002. URL http://www.sciencedirect.com/science/article/pii/S1389041708000429.

[59] Allen Newell. Unified theories of cognition. *Cambridge, MA: Harvard University*, 1990.

[60] David E. Kieras and David E. Meyer. An overview of the epic architecture for cognition and performance with application to human-computer interaction. *Hum.-Comput. Interact.*, 12(4):391–438, December 1997. ISSN 0737-0024. doi: 10.1207/s15327051hci1204_4. URL http://dx.doi.org/10.1207/s15327051hci1204_4.

[61] D E Meyer and D E Kieras. A computational theory of executive cognitive processes and multiple-task performance: Part 1. basic mechanisms. *Psychological review*, 104 (1):3, 1997.

[62] John Robert Anderson and Christian Lebiere. *The atomic components of thought*. Psychology Press, 1998.

[63] John R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. L. Qin. An integrated theory of the mind. *Psychological review*, 111(4):1036–1060, 2004. doi: 10.1037/0033-295x.111.4.1036.

[64] Z. Duric, W. D. Gray, R. Heishman, Fayin Li, Azriel Rosenfeld, M. J. Schoelles, C. Schunn, and H. Wechsler. Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction. *Proceedings of the IEEE*, 90 (7):1272–1289, Jul 2002. doi: 10.1109/JPROC.2002.801449.

[65] Thomas L. Griffiths, Nick Chater, Charles Kemp, Amy Perfors, and Joshua B. Tenenbaum. Probabilistic models of cognition: exploring representations and inductive biases. *Trends in Cognitive Sciences*, 14(8):357–364, 8 2010. doi: http://dx.doi.org/10.1016/j.tics.2010.05.004. URL http://www.sciencedirect.com/science/article/pii/S1364661310001129.

[66] Michael D. Lee. How cognitive modeling can benefit from hierarchical bayesian models. *Journal of Mathematical Psychology*, 55(1):1–7, 2 2011. doi: http://dx.doi.org/10.1016/j.jmp.2010.08.013. URL http://www.sciencedirect.com/science/article/pii/S0022249610001148.

[67] B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to- one tutoring. *Educational Researcher*, 13:4–16, 1984.

[68] J. H. Larkin and R. W. Chabay. *Computer-Assisted Instruction and intelligent Tutoring Systems: Shared Goals and Complementary Approaches*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1992.

[69] J. R. Carbonell. *Mixed-initiative man-computer instructional dialogues. Final report*. MIT Press, Cambridge, MA, 1970.

[70] J. R. Carbonell. AI in CAI: an artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4):190–202, 1970.

[71] Paul A. Kirschner, John Sweller, and Richard E. Clark. Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist*, 41(2):75–86, 2006.

[72] Caroline Di Bernardi Luft. Learning from feedback: The neural mechanisms of feedback processing facilitating better performance. *Behavioural Brain Research*, 261(0):356–368, 3 2014. doi: http://dx.doi.org/10.1016/j.bbr.2013.12.043. URL http://www.sciencedirect.com/science/article/pii/S0166432813007924.

[73] César Coll, María José Rochera, and Ines de Gispert. Supporting online collaborative learning in small groups: Teacher feedback on learning content, academic task and social participation. *Computers & Education*, 75(0):53–64, 6 2014. doi: http://dx.doi.org/10.1016/j.compedu.2014.01.015. URL http://www.sciencedirect.com/science/article/pii/S0360131514000335.

[74] Kaśka Porayska-Pomsta and Chris Mellish. Modelling human tutors' feedback to inform natural language interfaces for learning. *International Journal of Human-Computer Studies*, 71(6):703–724, 6 2013. URL http://www.sciencedirect.com/science/article/pii/S1071581913000219.

[75] Martin Dougiamas and Peter Taylor. Moodle: Using learning communities to create an open source course management system. In David Lassner and Carmel McNaught, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003*, pages 171–178, Honolulu, Hawaii, USA, 2003. AACE. URL http://www.editlib.org/p/13739.

[76] Lee Underwood. A brief history of search engines. WebReference.com, 2004. URL http://webreference.com/authoring/search_history. Web. Retrieved July 25, 2014.

[77] Wes Sonnenreich. A history of search engines. Wiley, 1997. URL http://www.wiley.com/legacy/compbooks/sonnenreich/history.html. Web. Retrieved July 25, 2014.

[78] Reginald L. Walker. Search engine case study: searching the web using genetic programming and mpi. *Parallel Computing*, 27(1–2):71–89, 1 2001. doi: http://dx.doi.org/10.1016/S0167-8191(00)00089-2. URL http://www.sciencedirect.com/science/article/pii/S0167819100000892.

[79] Georges E. Dupret and Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 331–338, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390392. URL http://doi.acm.org/10.1145/1390334.1390392.

[80] Ryen W. White and Dan Morris. Investigating the querying and browsing behavior of advanced search engine users. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 255–262, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277787. URL http://doi.acm.org/10.1145/1277741.1277787.

[81] Chuan Wang, Guiran Chang, Xingwei Wang, Yan Ma, and Haitao Ma. A user motivation model for web search engine. volume 2, pages 327–331, 08 2009. ISBN 978-0-7695-3745-0. URL http://doi.ieeecomputersociety.org/10.1109/HIS.2009.178.

[82] Maeve O'Brien and Mark T. Keane. Modeling user behavior using a search-engine. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, IUI '07, pages 357–360, New York, NY, USA, 2007. ACM. ISBN 1-59593-481-2. doi: 10.1145/1216295.1216366. URL http://doi.acm.org/10.1145/1216295.1216366.

[83] Johannes Fürnkranz, Eyke Hüllermeier, Filip Radlinski, Madhu Kurup, and Thorsten Joachims. *Evaluating Search Engine Relevance with Click-Based Metrics*, pages 337–361. Springer Berlin Heidelberg, 2011. URL http://dx.doi.org/10.1007/978-3-642-14125-6_16.

[84] Bruce Grossan. Search engines: What they are, how they work, and practical suggestions for getting the most out of them. WebReference.com, 1997. URL http://webreference.com/content/search/. Web. Retrieved July 25, 2014.

[85] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18):3825–3833, 2012. doi: http://dx.doi.org/10.1016/j.comnet.2012.10.007. URL http://www.sciencedirect.com/science/article/pii/S1389128612003611.

[86] Christopher L. Asplund, J. Jay Todd, Andy P. Snyder, and René Marois. A central role for the lateral prefrontal cortex in goal-directed and stimulus-driven attention. *Nature Neuroscience*, 13(4):507 – 512, 2010. URL http://search.ebscohost.com.proxy.bnl.lu/login.aspx?direct=true&db=mnh&AN=20208526&site=ehost-live&scope=site.

[87] Geoffrey F. Woodman and Steven J. Luck. Electrophysiological measurement of rapid shifts of attention during visual search. *Nature*, 400(6747):867–869, 08 1999. URL http://dx.doi.org/10.1038/23698.

[88] Alan D. Baddeley, Graham Hitch, and Gordon H. Bower. *Working Memory*, volume Volume 8, pages 47–89. Academic Press, 1974. doi: http://dx.doi.org/10.1016/S0079-7421(08)60452-1. URL http://www.sciencedirect.com/science/article/pii/S0079742108604521.

[89] Edward K. Vogel, Geoffrey F. Woodman, and Steven J. Luck. Storage of features, conjunctions, and objects in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1):92–114, 2001. doi: dx.doi.org/10.1037/0096-1523.27.1.92.

[90] Marvin M. Chun. Visual working memory as visual attention sustained internally over time. *Neuropsychologia*, 49(6):1407–1409, 5 2011. doi: http://dx.doi.org/10.1016/j.neuropsychologia.2011.01.029. URL http://www.sciencedirect.com/science/article/pii/S0028393211000340.

[91] Jan Theeuwes, Arthur F. Kramer, and David E. Irwin. Attention on our mind: The role of spatial attention in visual working memory. *Acta Psychologica*, 137 (2):248–251, 6 2011. doi: http://dx.doi.org/10.1016/j.actpsy.2010.06.011. URL http://www.sciencedirect.com/science/article/pii/S0001691810001186.

[92] Alan Baddeley. Exploring the central executive. *The Quarterly Journal of Experimental Psychology Section A*, 49(1):5–28, 1996. doi: 10.1080/713755608.