An Operational Driven Development Approach for Cognitive Agents in UAV Applications

Kai Reichel[1], Nico Hochgeschwender[1] and Holger Voos[2]

[1]ESG Elektroniksystem -und Logistik-GmbH, Livry-Gargan-Str. 6, D-82256 Fürstenfeldbruck, Germany
[2]University of Applied Sciences Ravensburg-Weingarten, P.O. Box 1261, D-88241 Weingarten, Germany
{kai.reichel,nico.hochgeschwender}@esg.de, voos@hs-weingarten.de

***Abstract*— Future applications of unmanned aerial vehicles (UAVs) require the operation of UAVs with a high level of autonomy. However, especially in military missions, the human UAV mission planners and operators should still be able to understand the behavior of the autonomous system. Therefore, this paper proposes the the application of cognitive agent technologies for the development of such systems. Several possible cognitive agent architectures are compared and assessed in order to detect the most suitable approach from an industrial perspective. Furthermore, a design procedure is being developed that supports the transition of the pure operational requirements and functional specification into a cognitive agent system: the *Op*erational driven development approach for *Cog*nitive Systems called *OpCog*. This procedure is then applied to an UAV SEAD (Suppression of Enemy Air Defence) scenario and some first lessons learned are discussed.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are investigated since decades and many different areas of both military and commercial applications have been considered. As UAVs are operated without any human pilot, they especially provide new applications in the military domain such as long endurance reconnaissance or Suppression of Enemy Air Defence (SEAD) missions. While most of the currently operated UAVs are mainly under remote control, is becomes obvious that future military UAV systems must provide a higher level of autonomy, see e.g. [9]. The human operator then takes over the role of a pure supervisor which is able to operate either single or even swarms or teams of UAVs. Although autonomous UAVs are an important area of research and many contributions can be found in the literature, see e.g. [3], [6] or [9], UAVs with such a high level of autonomy are not yet in operation.

The main reasons are the complexity of the required cognitive functions and the required safety, reliability and predictability of such an autonomous UAV system. In addition, there is still a gap between the view of the end user, i.e. the UAV mission planners and operators, and the viewpoint of the researcher that provides the technologies to implement the required cognitive functions. The mission planner has the task to define the operational requirements and the specific UAV functionalities like mapping, navigation, control, communication or coordination. He is mainly interested in a UAV system that fulfills the requirements and offers the required functionalities, and not in details of a technical implementation. On the other hand, many researchers seem to be mainly interested in providing some new cognitive architectures while neglecting questions of practical implementation and operational requirements.

The engineers and developers in industry which are responsible for the development of UAVs and all the subsystems like the onboard Mission Management System (MMS) then have to take these different aspects into account. They have to choose those technologies provided by research that allows the fulfillment of the operational requirements defined by the customer and the implementation of the core functionalities of the MMS. Hereby, these functionalities have to be designed and implemented in a way that also the UAV mission planners and operators should be able to understand the current behavior of the UAV during the mission. Furthermore, aspects of certification as well as time, budget, human resources and knowledge have to be considered too.

Therefore, the industrial development of autonomous UAV systems and the core functionalities of the MMS must be based on a suitable cognitive architecture and a design procedure that supports the transfer of the operational requirements and desired functionalities into a technical implementation in the UAV. One promising solution is the application of agent-based systems. Agents are especially suited to design cognitive autonomous systems, see e.g. [10] and also provide the extension to the multi-UAV case. In addition, numerous contributions related to UAV applications can be found in the literature, see e.g. [**?**], [3], [6] or [9], and methodologies, tools and implementation approaches as well as de-facto standards already exist.

In this paper, we focus on the industrial aspects of the design of an MMS for autonomous UAVs using agent technologies. For that purpose, we first describe a suitable application example of a SEAD mission that should be solved by autonomous UAVs. Hereby, the operational requirements from the UAV operator's point of view had to be derived. These requirements then lead to the identification of the core functionalities of the MMS. This MMS then should be implemented using a cognitive agent architecture, and hence a suitable solution from the industrial perspective had to be identified. Here, we decided to apply the BDI paradigm and to implement the core functionalities with the help of the COGNET architecture. Our experiences with that approach then led to the definition of a development process called OpCog: an operational driven development approach for cognitve agent systems. That approach is described here in more detail and some results and lessons learned during the project are presented.

II. UAV Application Scenario

During a UAV SEAD mission, data about possible targets like surface-to-air-missiles (SAM), including their current activities, capabilities and resources should be gathered. The mission task is the co-operative reconnaissance of the defined mission area by a heterogenous team of UAVs, i.e. a team of UAVs with different sensory capabilities. The mission is successfully accomplished if all targets in the mission area have been classified and localized.

A. Mission Profile

In general, a complete reconnaissance mission consists of several phases from "pre-flight mission planning" and "start from base" to "return to base" and "mission debriefing". In this paper we focus on the central mission phase, the aerial reconnaissance in the mission area, because it illustrates the operational complexity of a real UAV application scenario in the best way.

1) Participating Entities: The considered scenario includes the UAV operator in the ground control station as well as an undefined number of UAVs and targets. The targets can be classified according to their threat, size and mobility, ranging from large-size radar stations to small-size and highly dangerous, mobile SAM units. The UAVs are small and agile aircrafts which are already equipped with a flight control unit (FCU) and a highly integrated data link (HIDL) connecting them with the operator and among themselves. Moreover, each UAV carries a specific sensor equipment, either a radar or an electro-optical sensor. Both sensors are able to detect a target, whereas the radar sensor can also determine the exact position of a target and the electro-optical sensor can classify and hence identify a target. Therefore, a target classification and localization can only be achieved by merging both data.

2) Mission Phases: In order to improve the mission results it is reasonable to divide the mission in two phases, a coarse find fix and track (FFT) phase and a fine FFT phase. The goal of the FFT coarse phase is the detection of all targets in the mission area by segmenting the area in several sectors, allocate them to the UAVs and clearing them up individually by one UAV only. In the FFT fine mission phase, subteams are formed whose task is the localization and classification of the targets which have been detected during the previous phase. Hence, each team must comprise at least one UAV with radar sensor equipment and one UAV equipped with an electro-optical sensor. The allocation of sectors to UAVs as well as the targets to the subteams should be accomplished according to the optimal application of all available resources.

B. Operational Requirements

Operational requirements describe the needs on mission level which have to be met in order to succeed the task of the mission. They can also be referred to as the goals which have to be achieved or maintained during the mission course. The operational requirements can be classified on the one hand in those that are specific for each mission phase. On the other hand, general requirements exist which have to be fulfilled at any time during the mission course.

1) Mission Phase FFT Coarse: At the beginning of the mission the UAVs share information about their mission goal. If they realize that their goals can only be achieved by cooperation, they should build a team and thus having the goal of *Team Building*. if a team is built, the mission area has to be divided into parts and distributed among the team members, *Sector Distribution*. Furthermore, the sectors should be cleared up, *Sector Reconnaissance*, and the detected targets should be communicated within the team, *Communication of Detection Results*.

2) Mission Phase FFT Fine: Having finished the FFT coarse phase, the team should build subteams, *SubTeam Building* , composed of at least one UAV equipped with a radar sensor and one with an electro-optical sensor. The whole team must be able to allocate the targets among the different subteams, *Target Allocation*. In order to provide optimal mission execution the subteams shall compute a resource and threat minimizing flight path to cover all targets, *Optimized Path Planning*. The classification and localization of targets can only be achieved by a fusion of the different sensory results, *Target Data Fusion*. The outcome of the fusion shall be communicated in the team as well as to the operator to prevent target data loss in case of an UAV loss, *Target Data Communication*. As new threats could be identified hereby, the UAVs shall replan their flight path to minimize the threat risk, *Path Re-Planning*.

3) General: One general operational requirement is the reconfiguration of the team and the subteam, *Team Reconfiguration* and *Subteam Reconfiguration*. This indicates that the UAVs are able to identify the non-reachability of their mission goal with the current team or subteam configuration and hence rebuild the team and subteam. In addition, the UAVs shall end the mission and dissolve the team when either the mission goal has been accomplished or can not be achieved any longer, *Mission Ending*. Ensuring the flight safety, each UAV must avoid collisions with other aircrafts or the ground, *Collision Avoidance*, and provide adequate handling of its flight behavior by the help of a flight control unit (FCU), *UAV Guidance*.

III. Cognitive Agent Systems

A. Cognitive Agent Systems: A Survey

A cognitive agent system can be defined as *a technical system embedded in a complex environment, that gathers and processes information in order to act in and thereby alter the environment by own behavior. Herein, the information processing imitates the human cognitive behavior according* *to the aforementioned phenomena and characteristics,* i.e. as an agent that processes the information according to a model of human cognition. There are several reasons why the modeling of human cognition in such cognitive agent systems is useful: (1) the actions of cognitive agents should be more human-like and understandable to the people that need to interact with them, (2) the knowledge the agents need should be more readily obtainable from human experts in the same field of work and (3) it should be easier to analyze and debug the agent's internal reasoning and thought processes. Because of these properties, the application of cognitive agents is of special interest in industrial applications and led to our decision to use cognitive agents for the realization of autonomous UAVs in this project.

There are several candidate architectures based on human cognition and reasoning such as Soar [4], ACT-R [1], BDI [7] and COGNET [**?**]. While Soar and ACT-R are among the classical approaches of cognitive architectures and intensively described and discussed in the related literature, see e.g. [4] and [1], only the BDI and the COGNET architectures are described here in some detail.

BDI agents are based on the philosophical concepts of intentions, plans and practical reasoning developed by Bratman [2]. The BDI paradigm is described in detail in [7] and is one of most common approaches for the design of agents. The BDI architectures have their roots in the tradition of understanding practical reasoning and comprise two important processes: deciding what goals should be achieved (deliberation) and how to achieve these goals (means-ends reasoning) [10]. The beliefs represent the information the agent has about its current environment. The beliefs of an agent could be different from the actual state of the world, because the sensors may be imperfect.

Desires are the main constant goals of the agent. These give the state of the world in which the agent wishes to be, and must be consistent. The intentions are the commitments to some of these goals according to the current situation. In order to reach a goal, the agent chooses a plan out of a plan library. There may be more than one current plan, because an agent may be simultaneously working towards multiple (non-conflicting) goals. Once an agent forms an intention (and selects a plan) it is in some sense committed to that goal. It will continue executing the plan (or at least have an intention to execute it) until the goal is achieved, the goal becomes irrelevant, or it is impossible to proceed with that plan. A plan is a recipe to achieve a particular goal. It is a sequence of actions and/or sub-goals to achieve. If any step in the sequence fails, the plan itself will fail. One of the features of a BDI system is that when a plan fails, the agent will recover (if possible). It will try to find another way of achieving the goal, taking into account the fact that the world (and hence the agents beliefs) is changing [5].

According to [10], there are seven main components of a BDI agent to perform practical reasoning: (1) a set of current beliefs, (2) a belief revision function that takes the perceptions and current beliefs to generate a new set of beliefs, (3) an option generating function that determines the available options of the agent, i.e. the current desires, on the basis of the current beliefs and intentions, (4) a set of current options representing possible courses of actions, (5) a filter function which is the deliberation process to determine the agents intentions on the basis of the current beliefs, desires and intentions, (6) a set of current intentions the agent has committed to and (7) an action selection function which determines an action on the basis of the current intentions. For further details on the original BDI architecture we refer e.g. to [10], [7].

COGNET stands for "Cognition as a Network of Tasks" and is mainly based upon the work [**?**]. COGNET is also a cognitive architecture, but was developed especially for tactical decision making in military missions, characterized by real-time requirements, decision under non-predictable events and multi-tasking. The human information processing is modeled as the parallel execution of three mechanisms: perception, cognition and motor actions. Perception gathers information from the external world and makes it internally available as symbolic or semantic information stored in an extended working memory. A cognitive process analyzes and evaluates the symbolic representation of the world according to a goal catalog. If conflicts occur, a special conflict resolution takes place. As a result of cognition, the symbolic representation is altered and motor actions are generated to act in the real environment.

COGNET considers four main types of knowledge: declarative, procedural, perception and action knowledge. The declarative knowledge comprises the internal representation of the external world and is stored in form of a blackboard structure. Such a blackboard is organized in a hierarchical form in panels and sub-panels, each containing knowledge elements. Information processing in COGNET is understood as the activation and execution of procedural knowledge, which is organized in form of so called cognitive tasks. It is assumed that an overall mission must be accomplished which can be expressed as a network of such interdependent cognitive tasks. Some of these tasks have to be executed in parallel, some of them in a sequential manner. Each task comprises the context and the priority of the task, and the procedural knowledge how to reach the task goal. Each task contains goals that must be achieved to solve the task, the goals are also organized in a hierarchical form.

The activation of the tasks is done by a meta-cognitive trigger, that activates all tasks which are relevant in the actual context. The meta-cognitive trigger reacts to changes on the blackboard and also schedules the execution of the cognitive tasks in a way that only one task can be active at one time. If several tasks are relevant in a certain situation, the meta-cognitive trigger considers the priority of the tasks and triggers that task with the highest priority. Each task is formally described by a task definition and a task body. The task definition identifies the goal that has to be achieved and the trigger condition and priority. The task body comprises a hierarchy of sub-goals and operators. These operators can be action operators (to act externally), cognitive operators (to manipulate the declarative knowledge on the blackboard) and meta-cognitive operators (control the task execution).

Another form of knowledge in COGNET is perceptual knowledge. This perceptual knowledge is expressed as if-then rules (perceptual demons) that always run in parallel to the execution of the cognitive tasks. The perceptual demons modify the blackboard based on some triggering events in the environment. While COGNET is the conceptional framework, iGEN is the software toolkit that supports the development of cognitive agents based on COGNET. It allows a complete graphical definition of the blackboard, the cognitive tasks and the perceptual demons, see Fig. 1 for an example.

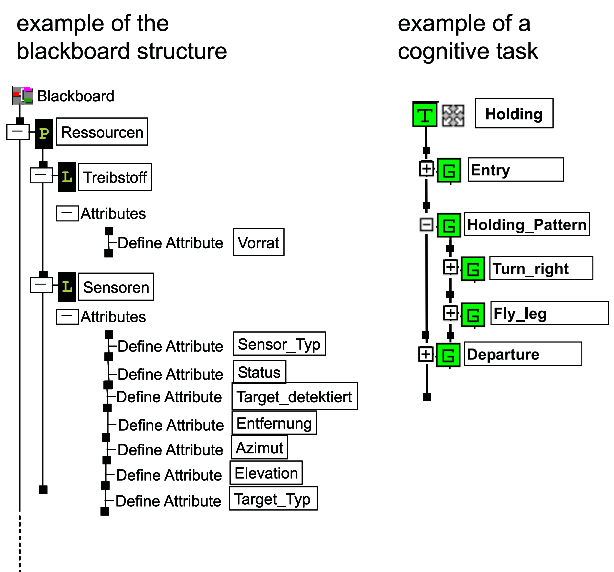

Fig. 1. Examples of blackboard structure and cognitive tasks in iGEN.

B. *Assessment of Cognitive Agent Systems*

In order to decide which of these paradigms fulfills the needs of the industrial developer of autonomous UAVs in the best way, all approaches have been compared and assessed with regard to some suitable evaluation criteria. These criteria are categorized as company-, theory- and application-specific criteria. *Company-specific* criteria comprise already available knowledge and human resources of the company as well as long-term development strategies. *Theory-specific assessment criteria* have been chosen as (1) complexity, (2) syntax, flexibility of the approach, (3) required theoretical knowledge to apply the approach, (4) degree of dissemination, (5) effort for knowledge extraction and implementation, (6) modularity, extendability and (7) realization of cooperative behavior. The *application-specific criteria* are (1) suitability for UAV scenarios, (2) reference projects in the UAV area, (3) requirements wrt practical (programming) knowledge, (4) available tools and documentation and (5) license fees.

Taking these criteria into account, an assessment led to the result that both Soar and ACT-R are not really suitable for the considered application domain. This is mainly caused by the rather complex syntax, the unflexible architecture and the missing tool support. In addition, these paradigms do not really support the formation and coordination of teams. The main advantage of the BDI paradigm is the fact that it is a well known approach supported by numerous tools. The theory itself is straightforward, but the implementation from a practical/industrial perspective is most often rather difficult. The definition of the beliefs, goals and plans most often has to be done with formal languages which are difficult to understand for the UAV planner/operator. The main advantage of the COGNET/iGEN approach is the graphical description and definition of all components of the architecture.

Therefore, the main result of the assessment was the idea to merge the BDI paradigm and the COGNET/iGEN implementation. The BDI paradigm is used as the underlying concept that models the cognitive behavior of the agent. That model of the cognitive agent is then mapped to the components used by COGNET/iGEN and the graphical description of knowledge, goals and plans. The same idea can also be found in [8], where the authors discussed the application of the iGEN toolkit for modeling cognitive BDI agents in a naval training scenario.

For the implementation of belief generation of the cognitive agent, various components are used. Part of the agents beliefs are not generated during runtime, but are predefined. Those beliefs are stored on the blackboard at the beginning of the mission. Beliefs about the external world are generated by the perceptual demons during runtime, which also transfer and integrate incoming information to the blackboard. The desires of the BDI paradigm can be interpreted as the full set of cognitive tasks, that become intentions if the tasks are triggered according to the current situation (i.e. fulfillment of precondition and priority). The hierarchy of sub-goals and operators in the cognitive tasks implements the plans that have to be followed in order to fulfill the cognitive tasks and achieve the goals. Therefore, a BDI-like cognitive agent can be implemented using the COGNET/iGEN approach without bigger problems. Thus, from an industrial perspective that approach provides a suitable and comparatively comfortable method to design cognitive agent systems. As the specific UAV domain has not been considered so far, the question remains how to transfer the operational requirements of a specific UAV mission into a cognitive agent implemented with COGNET / iGEN. The investigation of this problem led to the proposal of an integrated development procedure based on BDI and COGNET/iGEN.

IV. OpCog

In this section we present *OpCog* as an operational driven development approach for cognitve agent systems. As the acronym indicates, *Op* stands for "operational" and *Cog* for "cognitive". The *OpCog* development approach tries to bridge the gap between the operational requirements derived from the military user and the existing cognitive systems. During the development processes in former projects in the area of UAV applications, we identified three main

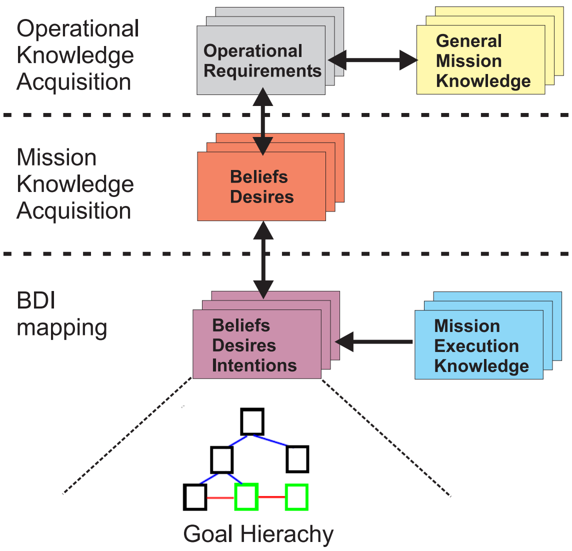

Fig. 2. The development approach.

stakeholders: domain experts, operators and developers. Domain experts like military users are interested in the overall fulfillment of their requirements according to standards and specifications, see section II. Operators are the real users of the system and know exactly the real mission course. The third group of stakeholders comprises the industrial developers which have to capture and transform the knowledge from operators and experts into a working UAV system. Our development approach consists of three development phases, see figure 2:

1. Operational Knowledge Acquisition: During the knowledge acquisition phase domain experts and developers emerge potential operational requirements like those described in section II-B. For that reason domain experts provide general mission knowledge e.g. military procedures and mission specific schedules. In general, domain experts have no deep understanding of cognitive technologies. Therefore the developers consult the domain experts in technology related questions which are important in the overall design of the cognitive system.

2. Mission Knowledge Acquisition: Based on the operational requirements the developers then have to derive the beliefs and desires of the cognitive system. Beliefs represent the knowledge which is required to accomplish the mission and can be categorized in a priori knowledge and knowledge which is generated and updated at runtime. Desires represent the goals which the agent wants to achieve. In our approach we distinguish between two types of goals: non-measurable and measurable goals called *abstract* or *real* goals, respectively. The two-fold distinction of goals is decoupled from the underlying cognitive system. Unlike [8] neither abstract nor real goals are interlocked to concrete beliefs. However, based on the operational requirements obtained in phase one, the developers are able to model the causal and hierarchical relation between abstract and real goals, see also Fig. 3.

3. BDI Mapping: In the third development phase the operators provide mission execution knowledge which is used to complete the BDI model of the cognitive agent. Note that up to now we derived only the beliefs and desires. According to the theory of the BDI paradigm, intentions are instanced desires at a certain point of time. Based on the provided mission execution knowledge, the developers derive the temporal relations between goals and their circumstances. Hence, result of the third development phase is a goal hierarchy provided temporal and causal relations, see also section 2 and Fig. 3.

After a revision of the goal hierarchy the developers start to implement and integrate the proposed system. The whole procedure is now explained in more detail taking the application example into account.

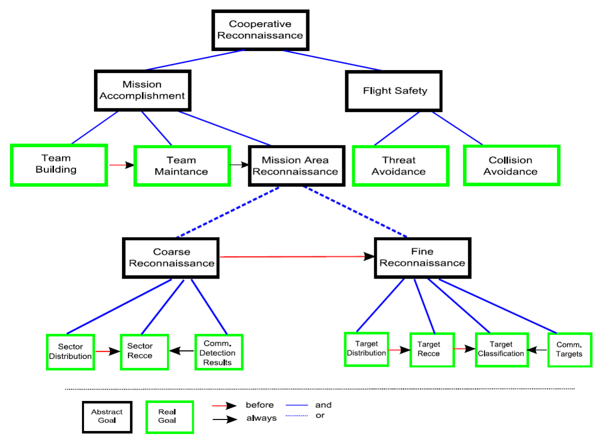

Fig. 3. The goal hierarchy.

The *OpCog* development approach has been applied to the mission scenario described in section II-A. Then, this formal system description using the BDI paradigm has been mapped to COGNET/iGen for the implementation.

First, the operational requirements have been identified together with the domain experts, see section II-B.

Second, from the operational requirements we derive the beliefs and goals needed to accomplish the mission. Furthermore, the causal relations between the goals are identified and used to setup the goal hierarchy for this mission (see Fig. 3). The goal hierarchy is based on an AND/OR goal graph where AND/OR links represent causal relations between goals. Thus, a goal depending on two lower level goals which are linked with the AND annotation can only be accomplished if the two lower level goals are accomplished. The two types of goals, real and abstract ones, are presented as black and green boxes in Fig. 3, respectively. They only differ with regard to the determination of their accomplishment. The accomplishment of real goals can be measured, e.g. *Team Building* $\rightarrow$ a team has been built or not, whereas the accomplishment of abstract goals has to be derived from lower level goals in the hierarchy, e.g. the accomplishment of *Flight Safety* can only be derived from the measurable accomplishment of *Threat Avoidance* and *Collision Avoidance*.

Third, using the mission execution knowledge extracted with the operator we setup the temporal relations between the goals on the same level in the goal hierarchy. Hereby, we use the relations '`BEFORE`' in case a goal has to be achieved once before another one and '`ALWAYS`' in case a goal has to be achieved and maintained before another goal can be pursued. So, the goal *Team Building* has to be achieved once before the goal *Team Maintenance* can be pursued. But if e.g. the goal *Team Maintenance* is no longer achieved the goal *Mission Area Reconnaissance* can not be pursued. This could happen if one UAV in the team is destroyed. It is obvious that the remaining UAVs have to rebuilt the team in order to achieve the overall mission goal.

Forth, after having modelled the behaviour specification of the UAVs, the BDI paradigm has to be mapped to the COGNET/iGEN architecture and toolset. The declarative part of the beliefs as well as the desires have been implemented using the blackboard structure. The reasoning cycle and the procedural part of the beliefs have been described by COGNET/iGEN cognitive tasks (CT). The COGNET/iGEN framework itself already includes a reasoning cycle based on the current priority and trigger conditions of CT. In order to fulfil the behaviour specification it is necessary to design an add-on reasoning cycle working only on the goal hierarchy. This add-on reasoning cycle has been implemented using a high priority goal evaluation CT. That goal evaluation CT examines the current achievement of the goals according to the causal and temporal relations between them. Based on the current achievement values of the goals and their relations the goals are prioritized. The add-on reasoning cycle is repeated at each update of the system. The great benefit of this approach is the generality of the add-on reasoning cycle. Therefore, it is completely independent of any goal hierarchy. That enables us to adapt the system rapidly to new desired behaviors.

The developed cognitive system has been integrated in a simulation framework providing the core functionalities of the SEAD mission like sensor simulation or flight dynamics simulation. Using this simulation framework, the cognitive system implementation and the *OpCog* development approach have been investigated leading to some first lessons learned.

V. Conclusions, Lessons Learned

This paper proposes the *OpCog* methodology that allows the mapping of operational requirements extracted from the domain experts to current cognitive agent approaches. Here, the BDI paradigm has been chosen for the description of the system behavior and COGNET/iGEN for the implementation, similar to [8]. In contrast to [8], our generic approach leads to a separation between behavior specification and actual execution. Thanks to the temporal and causal relations in the goal hierarchy one can determine (1) which goals must be achieved, (2) why these goals must be achieved and (3) when they have to be achieved. Moreover, as the behavior specification is visual (see Fig. 3) one can easily understand and refine it. This was very helpful during the incremental validation process carried out together with the domain expert and the operator.

During evaluation we simulated the SEAD mission using different test cases. The cases differed in parameters like e.g. number of available UAVs, sensor equipment, fuel resources, etc. Part of the evaluation was the input from a UAV operator who monitored the execution of the test cases. One result was the fact that the missions were generally executed in accordance to the operational requirements. However, if the number of UAVs exceeded six the operators were no longer able to maintain an overall situational awareness. This was mainly caused by the unsufficient man-machine-interface and hence future work should also consider the visualization of the complex behavior of a multiagent system.

Furthermore, the operator was impressed that he could change the behavior specifications with regard to the current mission goal. For example, he could easily change the temporal relation between *flight safety* and *mission accomplishment* in a way that mission accomplishment had always priority to flight safety. From the industrial point of view, another problem of the *OpCog* approach is the requirement that the developer still must have appropriate background knowledge in cognitive systems engineering in order to model a complex system. Therefore, further work must focus on a separation between the cognitive technology itself and the application. Moreover, future investigations should also consider the questions of validation and verification of such cognitive agent systems since this is essential for certification, see [12] for a perspective.

References

[1] R. Anderson and C. Lebiere. *The Atomic Components of Thought*. London: Lawrence Erlbaum Associates Publishers, 1998.

[2] M. Bratman. *Intentions, Plans and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.

[3] S. Karim and C. Heinze. Experiences with the design and implementation of an agent-based autonomous UAV controller. In *Proc. of the fourth international joint conference on Autonomous agents and multiagent systems AAMAS 05*, New York, NY, USA, 2005. ACM.

[4] A. Newell. *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press, 1990.

[5] E. Norling. Capturing the quake player: Using a BDI agent to model human behaviour. In *Proc. of the second international joint conference on Autonomous Agents and Multiagent Systems AAMAS 03*, 2003.

[6] M. Pechoucek, S. Thompson, and H. Voos. *Defence Industry Applications of Autonomous Agents and Multi-Agent-Systems*. Basel: Birkhauser Verlag, 2007.

[7] A. Rao and M. Georgeff. BDI-agents: from theory to practice. In *First Int. Conf. on Multi-Agent Systems*, 1995.

[8] W. A. van Doesburg, A. Heuvelink, and E. L. van den Broek. Tacop: A cognitive agent for a naval training simulation environment. In *Proc. of the fourth international joint conference on Autonomous agents and multiagent systems AAMAS 05*, 2005.

[9] H. Voos. Autonomous systems approach to UAVs. In *18th Bristol International Conference on Unmanned Air Vehicle Systems*, 2003.

[10] M. Wooldridge. *Reasoning about rational agents*. MIT Press, 2000.

[11] W. Zachary and J.-C. L. Mentec. A framework for developing intelligent agents based on human information processing architecture. In *1999 IASTED International Conference ASC*, 1999.

[12] N. Hochgeschwender and H. Voos. Verification of Autonomous Robotic Systems: A Perspective. In *IASTED International Conference of Robotics and Applications and Telematics (RA)*, 2007.