1

Agent-Based Distributed Resource Allocation in Continuous Dynamic Systems

Holger Voos
University of Applied Sciences Ravensburg-Weingarten
Germany

1. Introduction

The automation of complex large-scale systems is one of the most challenging tasks of modern control engineering. Such systems comprise a huge number of spatially distributed subsystems with both frequent and infrequent interactions, resulting in a complex overall behavior. In addition, numerous disturbances can occur leading to a high degree of uncertainty. The automation of such systems therefore calls for highly flexible automatic control systems that fulfill the following requirements:

- In accordance with the system under control, the overall control system should also consist of a number of spatially distributed local control systems, interconnected via suitable communication systems. In addition, also the control algorithms should be decentralized without any central control and the local control decisions are then coordinated to an overall consistent decision.
- The control functionality is implemented in the form of software which requires that state-of-the-art software engineering methods and techniques be employed, see [Bussmann, 2003].
- Because of the complex structure and the uncertainty of the system under control, the control algorithms must be robust against any model inaccuracies as well as disturbances during operation.
- It should be easy to maintain, to reconfigure or to extend the control system.
- The control system should also provide cognitive capabilities in order to realize the necessary complex decision making.

Taking these requirements into account, intelligent agents and multiagent systems reveal new strategies to design automation systems especially when considering large-scale distributed applications [Unland, 2003]. Agents can be defined as computer or software systems which are situated in some environment and able to perform flexible autonomous action in order to meet their design objectives [Unland, 2003], [Jennings, 1998]. Furthermore, agents can be pro-active and social, which enables them to interact with each other to form multiagent systems. In a multiagent system, the agents coordinate their behavior and solve problems in a distributed fashion without global control and only local and limited resources and information. There are first promising industrial applications of multiagent systems for the control of manufacturing, logistics, traffic or multi-robot systems, see e.g. [Weiss, 1999], [Colombo, 2004], [Kluegl, 2005], [AAMAS, 2008] for a survey.

Source: Multiagent Systems, Book edited by: Salman Ahmed and Mohd Noh Karsiti,
ISBN 978-3-902613-51-6, pp. 426, February 2009, I-Tech, Vienna, Austria

One special subproblem during the control of large-scale technical systems is automatic resource allocation. In manufacturing systems for instance, the capacities of the machines are resources that have to be allocated to the production orders. In logistic systems, trucks and transportation lines as resources must be allocated to the transportation demands of the customers. However, automatic resource allocation is not limited to the control of manufacturing and logistic systems but also occurs in computer and communication networks, production plants, traffic and transportation systems, energy networks or in building automation. In a distributed control system, the task of automatic resource allocation is preferably also performed in a distributed manner, leading to distributed resource allocation. Therefore it is very interesting to note that the previously proposed multiagent systems are especially suited to solve the problem of distributed resource allocation, see [Weiss, 1999], [Unland, 2003].

However, most of the resource allocation problems solved so far with the help of multiagent systems are static problems where the allocations do not depend on time. Many resource allocation problems of practical interest can be solved using these static considerations, even in discrete-event systems like manufacturing or logistic systems. In these cases, the necessary allocation is computed based on the current state of the system and that allocation is maintained until some new events or changes of the states occur. Unfortunately, problems especially in highly dynamic environments cannot be addressed by this pure static approach since the allocations, i.e. the decision variables, depend on time and previous states of the considered system. Hence, continuous-time allocation trajectories must be computed and this processing must be performed in real-time. These problems are hardly considered in the relevant agent literature and if, most often only discrete-event systems are considered. Therefore, this work focuses on dynamic resource allocation problems especially in continuous systems.

The design of a multiagent system for distributed resource allocation mainly comprises the design of the local capabilities of the single agents and the interaction mechanisms that makes them find the best or at least a feasible allocation without any central control. Many possible interaction mechanisms can be found in the literature, see e.g. [Weiss, 1999], [Sandholm, 1999] for an overview. This work proposes a more formal approach, where the decision process of resource allocation is expressed as an optimization problem under certain constraints. This optimization problem can be solved in a distributed fashion using multiple agents that act as local optimizers and coordinate their local solutions to an overall consistent solution, see also [Voos, 2003], [Voos, 2006], [Voos, 2007]. One special formulation of this optimization problem leads to an analogy with economic markets [Clearwater, 1996] and to so called market-based interaction mechanisms, see e.g. [Clearwater, 1996], [Voos, 2003]. Herein, supplies and demands of the resources are defined which are exchanged by agents and balanced using a virtual price. While the general approach of such a market-based resource allocation has already been investigated in the literature, this work adds two important contributions.

First, the market-based interaction mechanism is adapted here to a much more general class of resource allocation problems, extending this approach to more applications of practical interest. In addition, the method is further extended to cover resource allocation in dynamic systems. The corresponding mathematical formulation leads to dynamic optimization problems that could not be addressed so far by market-based algorithms. In the proposed solution, the agents calculate and negotiate complete supply and demand trajectories using model-based predictions which also leads to the calculation of a price trajectory. This novel approach does not only consider the dynamic behavior of the distributed system but also combines control tasks and resource allocation in a very consistent way. The agents are designed as two-level entities: while the low-level functions are responsible for the real-time allocation of the resources in the form of closed-loop feedback control, the high-level functionalities realize the deliberative capabilities such as long-term planning and negotiation of the resource allocations. The solutions are finally applied to a number of technical applications for proof of concept.

2. Resource allocation problems

2.1 Resource allocation in technical systems

In a technical environment, resources are all means that enable the operation of a technical process. Therefore, resources could be energy, information, materials, capacities of plants, machines etc. Therefore, if we consider a large production plant for instance, many examples of resource allocation problems can be found, see e.g. Fig. 1.

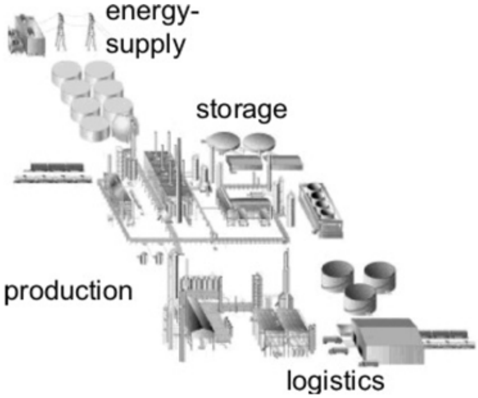

Fig. 1. Examples of resource allocation problems in a production plant.

It is characteristic that resources are always limited and quantities of a resource are always non-negative numbers. In a technical system, there are in general subsystems that offer resources and other subsystems (or entities) that need resources. Considering a communication network as an example, the channels offer the resource “bandwidth” which is used by the active communication connections. In a shop-job manufacturing system, the machines offer the production capacities as a resource which are then used by the production orders. In many technical systems however, the resources are not only used by one single entity but there are several of them that have to share limited quantities of the resources. Therefore, a sufficient quantity of each available resource must be allocated to each entity. Most often, several alternatives for that allocation exist which means that resource allocation is a decision process in general.

The resource allocation problem considered so far is static, i.e. the solution doesn’t depend on time. In a technical realization, the allocation is calculated and maintained until a disturbance of that state occurs (e.g. if a new entity appears that also needs resources). In this case, the allocation is recalculated based on the new situation. Many problems of

practical interest could be solved like this, e.g. in computer or communication networks or production planning in manufacturing systems [Colombo, 2004], [Weiss, 1999].
One special resource allocation problem however appears in dynamic systems with continuous state variables, where the allocation of a certain amount of a resource also depends on the internal state of the system. An example is the allocation of heating energy as a resource to the single production processes e.g. in a chemical plant, where the allocation depends on the current temperatures in the different tanks. The temperature, however, can be described by a differential equation and also depends on previous allocations of heating energy. Thus allocation now depends on time and resource allocation is a dynamic continuous process. In addition, resource allocation then turns out to be a control problem in this case. The resource, i.e. the heating energy, is the controlled input variable of the considered process. The control goal is to maintain a certain temperature in the tank. Therefore the current temperature is measured, compared with the required setpoint temperature and the current allocation of the resource is adapted with regard to the deviation from this setpoint.
It is obvious that the dynamic behavior of the technical subsystems that need resources (e.g. the chemical reaction in the tank) now plays an important role during the solution of the allocation problem. One possible approach which will be derived in this contribution is a direct combination of the mathematical description of the resource allocation problem and the models of the dynamic behavior of the subsystems. Together with a market-based approach using a multiagent system, it leads to an interaction and control scheme that is similar to a model-predictive control algorithm. A second approach which will also be derived in this chapter is based upon a separation of the resource allocation process on the one side and the control problem on the other side. Here, while dealing with dynamic resource allocation, it is sufficient to work with a static mathematical formulation of the resource allocation problem and a static market-based interaction scheme.

2.2 Mathematical formulation of resource allocation

In many technical systems, the resources are not only used by one single entity but there is a number of entities that have to share limited quantities of the resources. Therefore, a sufficient quantity of each available resource must be allocated to each entity. Most often, several alternatives for that allocation exist which means that resource allocation is a decision process in general. Here we assume that we are interested in the best possible allocation based on the available information which leads to the formulation as an optimization problem, see also [Ibaraki, 1988].
In the following we assume that a number of L resources exists that have to be allocated to I entities. The allocation to entity $i \in \{1, \dots, I\}$ can be expressed as an allocation vector $\boldsymbol{r}_i \in \mathbb{R}_+^L$ where r_{ij} denotes the allocated amount of resource j. If J is a suitable objective function that judges the overall allocation and if there is an overall maximum available amount of each resource given as the vector $\boldsymbol{r} \in \mathbb{R}_+^L$, the resource allocation problem can be expressed as follows:

$$
\begin{aligned}
& \min_{\{\boldsymbol{r}_i, \forall i\}} \quad J(\boldsymbol{r}_1, \dots, \boldsymbol{r}_I) \\
\text{s.t.} \quad & \boldsymbol{C} \cdot \boldsymbol{r}_i = \boldsymbol{d}_i \quad \forall i \\
& \boldsymbol{r}_i \geq \boldsymbol{0} \quad \forall i \\
& \sum_{i=1}^{I} \boldsymbol{r}_i = \boldsymbol{r}
\end{aligned}
\tag{1}
$$

The best allocation is that set of non-negative allocation vectors $\{\boldsymbol{r}_i, \forall \boldsymbol{i}\}$ that minimizes J while the overall sum of all allocations must be equal to the overall amount of the available resources. The equalities $\boldsymbol{C} \cdot \boldsymbol{r}_i = \boldsymbol{d}_i$ could be used to define some topological constraints where the topology of the system is described by a graph with incidence matrix $\boldsymbol{C}$ (see the example in the last section for further explanation).
Distributed resource allocation however means that there is no central control but a number of distributed agents assigned to the entities $i = 1, \ldots, I$ which communicate in order to coordinate the allocation in a distributed manner. Therefore, the description (1) of the resource allocation problem as one single optimization problem is not suitable. If we assume that an objective function $J_i(\boldsymbol{r}_i)$ for each entity i exists that measures the efficiency of the local resource allocation to that entity, the overall objective function J in (1) can be replaced by

$$J = \sum_{i=1}^{I} J_i(\boldsymbol{r}_i) \tag{2}$$

The resource allocation problem considered so far is a static description, i.e. the solution doesn't depend on time. In a technical realization, the allocation is calculated and maintained until a disturbance of that state occurs (e.g. if a new entity appears that also needs resources). In this case, the allocation is recalculated based on the new situation. Many problems of practical interest could be solved like this, e.g. in computer or communication networks or production planning in manufacturing systems [Weiss, 1999], [Colombo, 2004].
One special resource allocation problem however appears in dynamic systems with continuous state variables, where the allocation of a certain amount of a resource also depends on the internal state of the system. An example is the allocation of heating energy as a resource to the single rooms in a building where the allocation depends on the current temperatures in the rooms. The temperature, however, can be described by a differential equation and also depends on previous allocations of heating energy. Thus allocation now depends on time and resource allocation is a dynamic continuous process. A possible mathematical formulation of that problem will be derived in the following.
We assume that the I entities that need resources are dynamic systems, here described by a discrete-time state variable model with the vector $\boldsymbol{x}_i(k)$ as the vector of continuous state variables at instant k. In addition we assume that the states also depend on the allocated amount of the resources, where $\boldsymbol{r}_i(k)$ is the allocation to entity i at instant k. Hence the system i is described by the difference equation

$$\boldsymbol{x}_i(k+1) = \boldsymbol{F}_i(\boldsymbol{x}_i(k), \boldsymbol{r}_i(k)) \quad , \quad \boldsymbol{x}_i(0) = \boldsymbol{x}_{i0} \tag{3}$$

and we additionally assume that the full vector of state variables can be measured or at least estimated. Now we have to consider the problem to allocate resources over a certain period of time comprising K time steps. The allocation to system i is no longer a single allocation vector but a trajectory of allocations given by $\{\boldsymbol{r}_i(0), \boldsymbol{r}_i(1), \ldots, \boldsymbol{r}_i(K-1)\}$. The objective function is now a function of the trajectory of allocations and the trajectory of the states. The resource allocation problem is given by

$$
\begin{aligned}
&\min_{\{\boldsymbol{r}_i(k),\forall i,k\}} \quad \sum_{i=1}^{I} \left(J_i(\boldsymbol{r}_i(0),\boldsymbol{x}_i(1),\ldots,\boldsymbol{r}_i(K-1),\boldsymbol{x}_i(K))\right) \\
\text{s.t.} \quad & \boldsymbol{x}_i(k+1) = \boldsymbol{F}_i(\boldsymbol{x}_i(k),\boldsymbol{r}_i(k)) \quad , \quad \boldsymbol{x}_i(0) = \boldsymbol{x}_{i0} \quad \forall i,k \\
& \boldsymbol{C}\cdot\boldsymbol{r}_i(k) = \boldsymbol{d}_i \quad \forall i,k \\
& \boldsymbol{r}_i(k) \geq \boldsymbol{0} \quad \forall i,k \\
& \sum_{i=1}^{I} \boldsymbol{r}_i(k) = \boldsymbol{r}(k) \quad \forall k
\end{aligned}
\tag{4}
$$

In order to apply agent-based resource allocation, we need some methodologies how to solve problems like (1) or (4) in a distributed fashion with communicating agents.

3. Agent-based resource allocation

3.1 Analogy between economies and resource allocation problems

The problem of distributed resource allocation has been addressed from the beginning of agent-based research and application. Here algorithms or methodologies have been developed that especially take into account the decentralized system structure of multiagent systems and their ability to communicate and coordinate. Well known methods of multiagent based resource allocation comprises blackboard structures or auction-like algorithms, see [Weiss, 1999], [Sandholm, 1999] for a summary. One method which will be investigated here is based on economic markets, since resource allocation is also a basic problem in human societies.

An abstract mathematical model of an idealized economic system[Debreu, 1959] consists of a certain number of commodities, agents and a price system. A commodity can either be a service or a good; any quantity of a commodity is a positive real number. Because of the limitation of the commodities, each is associated with a price while all prices together form the price system. The agents can either be consumers or producers and the task of an agent is to make the decision on a quantity of his input or output for each commodity. Each producer chooses his supply based on his production factors and has the objective of profit maximization. The consumers in the economy choose their demands, characterized by their choice criterions or preferences and certain constraints, mainly the limited wealth. In these models the consumer will always choose that demand he prefers the most and which is feasible by the wealth constraints.

On the market, overall supply and demand then has to be balanced by adjusting the price of the commodities: if the overall demand exceeds the overall supply, the prices are increased and vice versa. Under some strict conditions concerning the preferences and the production factors, such an economy can reach a competitive equilibrium where overall supply equals overall demand and where each consumer maximizes its preferences. Because of the principal and mathematical analogy between the distributed resource allocation problem and this model of an economic market, the idea of a market-based solution led to a number of solutions in agent-based distributed resource allocation, see [Clearwater, 1996], [Voos, 2003] for a first summary of applications.

One main drawback of this economic model are the constraints regarding the preferences of the consumers, most often expressed by so-called utility functions. To guarantee the existence of the competitive equilibrium, the utility of the consumer must be described by monotone, quasi-concave and strictly increasing functions. That means that the utility is increasing with increasing amounts of allocated commodities (”more is preferred”). The simple price adjustment algorithm called tâtonnement which is performed by the agents is only guaranteed to converge if these conditions hold. However, this may not be the case in technical systems, which causes problems for the direct adaption of the economic model to real technical applications. In addition, problems like (4) where additional variables such as the state variables appear in the objective function are not at all addressed in economic models. Therefore, from a mathematical point of view we have to consider two problems: 1.) how can problems of the form (1) be solved in a distributed market-based fashion even if the objective functions do not fulfill the mentioned strict conditions and 2.) how can we address dynamic problems like (4) by these market-based algorithms.

3.2 Market-based resource allocation with general objective functions

Now we consider (1) under the relaxed assumption that the objective functions $J_i(\boldsymbol{r}_i)$ are not strictly increasing, but strictly convex functions. The Lagrangian of optimization problem (1) is

$$\mathscr{L} = \sum_{i=1}^{I} J_i(\boldsymbol{r}_i) + \sum_{i=1}^{I} \boldsymbol{\lambda}_i^T (\boldsymbol{C}\boldsymbol{r}_i - \boldsymbol{d}_i) + \boldsymbol{p}^T \cdot (\sum_{i=1}^{I} \boldsymbol{r}_i - \boldsymbol{r}) - \sum_{i=1}^{I} \boldsymbol{\mu}_i^T \cdot \boldsymbol{r}_i \tag{5}$$

The conditions for optimality (which are necessary and sufficient for the existence of an optimum in the case of strictly convex functions J_i) are:

$$\begin{aligned}
&\frac{dJ_i(\boldsymbol{r}_i)}{d\boldsymbol{r}_i}\Big|_{(\boldsymbol{r}_i^*)} + \boldsymbol{C}^T \cdot \boldsymbol{\lambda}_i^* + \boldsymbol{p}^* - \boldsymbol{\mu}_i^* = \boldsymbol{0} \quad \forall i \\
&\boldsymbol{C}\boldsymbol{r}_i^* - \boldsymbol{d}_i = \boldsymbol{0} \quad \forall i \\
&-\boldsymbol{r}_i^* \leq \boldsymbol{0} \quad \forall i \\
&\boldsymbol{r}_i^{*T} \cdot \boldsymbol{\mu}_i^* = 0 \quad \forall i \\
&\boldsymbol{\mu}_i^* \geq \boldsymbol{0} \quad \forall i \\
&\sum_{i=1}^{I} \boldsymbol{r}_i^* - \boldsymbol{r} = \boldsymbol{0}
\end{aligned} \tag{6}$$

It is obvious from (6) that this optimization problem could also be solved as follows: for a given Lagrange multiplier $\boldsymbol{p}$ that is associated with the balancing equality condition, (6) can be decomposed into I independent optimization problems of the form

$$\begin{aligned}
&\min_{\{\boldsymbol{r}_i\}} \quad J_i(\boldsymbol{r}_i) + \boldsymbol{p}^T \cdot \boldsymbol{r}_i \\
\text{s.t.} \quad &\boldsymbol{C} \cdot \boldsymbol{r}_i = \boldsymbol{d}_i \\
&\boldsymbol{r}_i \geq \boldsymbol{0}
\end{aligned} \tag{7}$$

That means that each agent i tries to solve its own optimization problem for a given parameter $\boldsymbol{p}$ that leads to the allocation $\boldsymbol{r}_i^*(\boldsymbol{p})$. All of these single solutions together form the function

$$z(\boldsymbol{p}) = \sum_{i=1}^{I} \boldsymbol{r}_i^*(\boldsymbol{p}) - \boldsymbol{r} \tag{8}$$

This equation then can be used to find the optimal parameter $\boldsymbol{p}^*$ by the evaluation of

$$z(\boldsymbol{p}^*) = \boldsymbol{0} \tag{9}$$

and therefore also leads to the fulfillment of the last equality condition in (6). Hence $\boldsymbol{p}$ can be interpreted as a price vector that is used to balance overall supply and demand. Each agent i tries to optimize its allocation while minimizing the costs $\boldsymbol{p}^T \cdot \boldsymbol{r}_i$ of its allocation for a given current price. This is done by the single agents independent from each other which results in I demands that all depend on the current price (most often, however, not in the form of an explicit function). On the common market, the overall demand and the overall supply (here only the fixed supply $\boldsymbol{r}$) must be balanced by adjusting the price vector in the right way (market clearing).

This solution process can be executed in an iterative way: starting with a first price vector, all customers calculate their allocation by solving their own independent allocation problems. These allocations depend on the current price which in general cannot be given as an explicit function. Nevertheless, all current demand values are taken together and compared with the overall supply. If overall supply and demand are not equal, i.e. $z(\boldsymbol{p})\neq 0$, the price vector has to be adjusted in a suitable way. The price is distributed back to all agents that adapt their demands to that new price and so on. The basic problem is to adapt the price vector in a way that the overall process converges towards $z(\boldsymbol{p}) = 0$. In economic theory this iterative process is called tâtonnement and adjusts the price in the way

$$\boldsymbol{p}(\kappa+1) = \boldsymbol{p}(\kappa) + \boldsymbol{K} \cdot z(\boldsymbol{p}(\kappa)) \tag{10}$$

where κ is the index of the iteration steps and $\boldsymbol{K}$ is a suitable but constant matrix. If the conditions concerning the utility functions (e.g. strictly increasing) are fulfilled, it can be shown that a pure diagonal matrix $\boldsymbol{K}$ with elements $K_{ii} > 0$ on the main diagonal leads to convergence.

Since these conditions are no longer fulfilled here, a new algorithm for the iterative solution of (9) must be found. One algorithm that does not require the calculation of the Jacobian matrix is the algorithm of Broyden [Stoer, 1993]. This algorithm approximates the Jacobian matrix and works as follows:

$$\begin{aligned}
\boldsymbol{p}(\kappa+1) &= \boldsymbol{p}(\kappa) - \varphi(\kappa) \cdot \boldsymbol{d}(\kappa) \\
\boldsymbol{d}(\kappa) &= \boldsymbol{\Pi}^{-1}(\kappa) \cdot z(\boldsymbol{p}(\kappa)) \\
\boldsymbol{\delta}_1(\kappa) &= -\varphi(\kappa) \cdot \boldsymbol{d}(\kappa) \\
\boldsymbol{\delta}_2(\kappa) &= z(\boldsymbol{p}(\kappa+1)) - z(\boldsymbol{p}(\kappa)) \\
\boldsymbol{\Pi}(\kappa+1) &= \boldsymbol{\Pi}(\kappa) + \\
&\quad \frac{1}{\boldsymbol{\delta}_1^T(\kappa) \cdot \boldsymbol{\delta}_1(\kappa)} \cdot \left(\boldsymbol{\delta}_2(\kappa) - \boldsymbol{\Pi}(\kappa)\boldsymbol{\delta}_1(\kappa)\right) \boldsymbol{\delta}_1^T(\kappa)
\end{aligned} \tag{11}$$

The parameter $\varphi(\kappa)$ must be adjusted by optimization but can also be set to a constant value. It can be proven that this algorithm converges, see [Stoer, 1993]. This provides the mathematical basis for a first implementation of a multiagent market-based resource allocation.

3.3 Agent-based resource allocation

The first step during the engineering of a multiagent system is the derivation of a model with the help of a suitable modelling approach. One possibility to obtain such models is the application of the UML (Unified Modeling Language). However, the original version of the UML doesn't offer suitable constructs for the modelling of agent-based systems. Therefore, the UML was extended with some agent-specific constructs and diagrams leading to the so called AML (Agent Modelling Language), see [Cervenka, 2007]. Fig. 2 depicts a first AML model of the distributed resource allocation problem if we intend to solve it with the market-based approach.

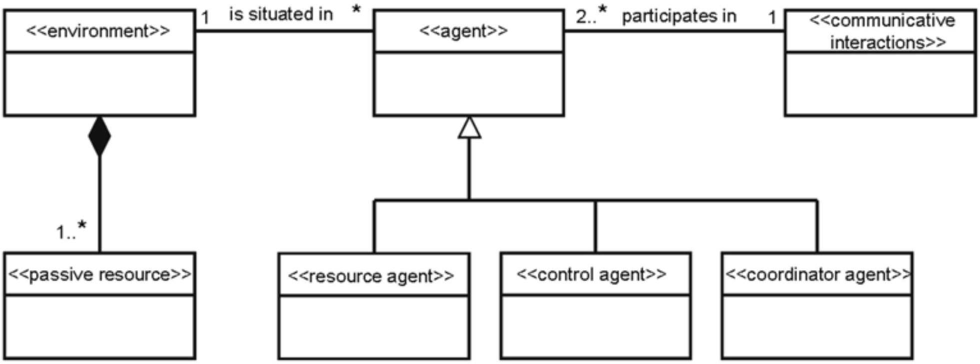

Fig. 2. AML model of the market-based resource allocation.

Both the resources that have to be allocated and the agents are situated in the environment. Here we distinguish between passive resources or resources with assigned resource agents. Resource agents however are only necessary if the (overall) amount of the resources can be influenced and therefore a further decision would be necessary. In the general case, the amount of the resources are fixed and no resource agents have to be assigned to them. The group of the agents itself can be further categorized into control agents and coordinator agents. The agents are interacting with the help of a suitable communicative interaction protocol which will be derived in the following.

The overall market-based resource allocation requires the assignment of one single control agent i to each of the I entities or subsystems that need resources and the definition of one coordinator agent. Then

1. The iteration index is set to $\kappa=0$. The coordinator agent chooses a start price $\boldsymbol{p}(\kappa=0)$ and a start matrix $\Pi(\kappa = 0)$. This price is transmitted to all other control agents via a communication network.
2. All control agents $i = 1, \ldots, I$ solve their local resource allocation problem (7). The result is a demand $\boldsymbol{r}_i^*(\kappa)$ which is locally optimal under the current price vector. All demands are transmitted to the coordinator agent.
3. The coordinator agent computes the current value of $z(\boldsymbol{p}(\kappa)$. If $z(\boldsymbol{p}(\kappa))=0$, the market is cleared and the allocation can be realized as calculated. Otherwise, the iteration index

κ is incremented by one and the coordinator agent calculates the new price vector $\boldsymbol{p}(\kappa+1)$ according to (11). The new price vector is distributed to the control agents and the algorithm proceeds with step two.

This interaction is shown in Fig. 3 in the form of an AML sequence diagram[Cervenka, 2007], using the multi-lifeline element for the control agents.

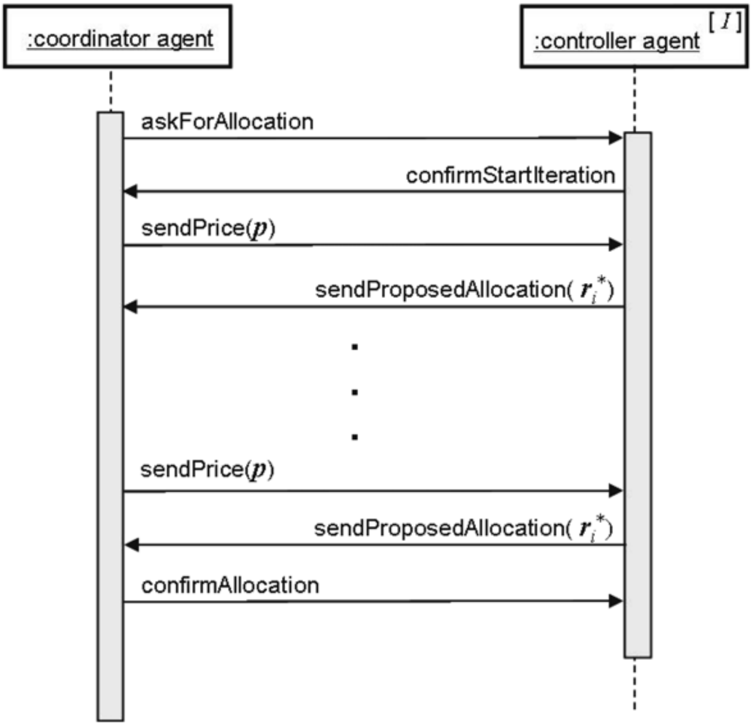

Fig. 3. AML sequence diagram of the interaction.

The only limitation we have so far is the condition of strict convexity of all objective functions $J_i(\boldsymbol{r}_i)$ of the single agents. It is no problem to generalizes this approach further by also adding some suppliers in the system and work with a varying and not fixed overall amount of the available resources. In this case, each of such resources are accompanied with a resource agent that is then also included in the interaction. The next problem that we have to address is the problem of resource allocation to dynamic systems.

3.4 Market-based resource allocation to dynamic systems

Now we assume that resources have to be allocated to a number of I dynamic subsystems, where each subsystem i is described by the discrete-time state variable model (3). Herein, the allocated resources are the inputs of the system. In many cases however, the subsystems can be described by a linear state variable model (which can be obtained by linearization):

$$\boldsymbol{x}_i(k+1) = \boldsymbol{A}_i\boldsymbol{x}_i(k) + \boldsymbol{B}_i\boldsymbol{r}_i(k) \quad , \quad \boldsymbol{x}_i(0) = \boldsymbol{x}_{i0} \tag{12}$$

A first approach for market-based dynamic resource allocation directly uses these dynamic models of the subsystems to form the objective functions as given in (4). Since we are now interested in trajectories of the state variables and the resource allocations, we express trajectories over K time steps in the form of vectors as

$$\boldsymbol{x}_i^T = (\boldsymbol{x}_i(1),\ldots,\boldsymbol{x}_i(K)) \quad , \quad \boldsymbol{r}_i^T = (\boldsymbol{r}_i(0),\ldots,\boldsymbol{r}_i(K-1)) \tag{13}$$

With the help of the state variable model (12), the connection between the state variable trajectory and the resource trajectory is given by

$$\boldsymbol{x}_i = \tilde{\boldsymbol{A}}_i\boldsymbol{x}_i + \tilde{\boldsymbol{B}}_i\boldsymbol{r}_i + \tilde{\boldsymbol{a}}_i \tag{14}$$

with

$$\tilde{\boldsymbol{A}}_i = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{A}_i & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ & \ddots & & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{A}_i & \boldsymbol{0} \end{pmatrix}, \tilde{\boldsymbol{B}}_i = \begin{pmatrix} \boldsymbol{B}_i & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_i & & \boldsymbol{0} \\ & & \ddots & \\ \boldsymbol{0} & \cdots & & \boldsymbol{B}_i \end{pmatrix} \tag{15}$$

and $\tilde{\boldsymbol{a}}_i^T = (\boldsymbol{A}_i\boldsymbol{x}_{i0}\,\boldsymbol{0}\,\cdots\,\boldsymbol{0})$. With the identity matrix $\boldsymbol{I}$ of a suitable size, (14) can then be reformulated to express the trajectory of the state variables as a linear transformation of the trajectory of the resources:

$$\boldsymbol{x}_i = (\boldsymbol{I}-\tilde{\boldsymbol{A}}_i)^{-1}\cdot\tilde{\boldsymbol{B}}_i\cdot\boldsymbol{r}_i + (\boldsymbol{I}-\tilde{\boldsymbol{A}}_i)^{-1}\cdot\tilde{\boldsymbol{a}}_i = \boldsymbol{\Psi}_i\boldsymbol{r}_i + \boldsymbol{c}_i \tag{16}$$

As in optimal control, the goal of a single agent i should be the minimization of the difference between the current states and required states $\boldsymbol{x}_{id}$ and the minimization of the resource consumption over a certain time interval. Since $\boldsymbol{x}_i$ and $\boldsymbol{r}_i$ are the trajectories of the states and allocations, the definition of the vector $\boldsymbol{x}_{id}$ as a vector of the trajectory of the required states and the diagonal weighting matrices $\boldsymbol{Q}_{i1}$ and $\boldsymbol{Q}_{i2}$ of suitable size yields the objective function of agent i

$$\begin{aligned} J_i(\boldsymbol{x}_i,\boldsymbol{r}_i) &= (\boldsymbol{x}_i-\boldsymbol{x}_{id})^T\boldsymbol{Q}_{i1}(\boldsymbol{x}_i-\boldsymbol{x}_{id}) + \boldsymbol{r}_i^T\boldsymbol{Q}_{i2}\boldsymbol{r}_i \\ &= \boldsymbol{x}_i^T\boldsymbol{Q}_{i1}\boldsymbol{x}_i - 2\boldsymbol{x}_{id}^T\boldsymbol{Q}_{i1}\boldsymbol{x}_i + \boldsymbol{x}_{id}^T\boldsymbol{Q}_{i1}\boldsymbol{x}_{id} + \boldsymbol{r}_i^T\boldsymbol{Q}_{i2}\boldsymbol{r}_i \end{aligned} \tag{17}$$

Now (16) can be inserted in (17) which results in

$$J_i(\boldsymbol{r}_i) = \boldsymbol{r}_i^T\boldsymbol{\Gamma}_i\boldsymbol{r}_i + \boldsymbol{h}_i^T\boldsymbol{r}_i + \alpha_i \tag{18}$$

with

$$\begin{aligned} \boldsymbol{\Gamma}_i &= \boldsymbol{\Psi}_i^T\boldsymbol{Q}_{i1}\boldsymbol{\Psi}_i + \boldsymbol{Q}_{i2} \\ \boldsymbol{h}_i^T &= 2\boldsymbol{c}_i^T\boldsymbol{Q}_{i1}\boldsymbol{\Psi}_i - 2\boldsymbol{x}_{id}^T\boldsymbol{Q}_{i1}\boldsymbol{\Psi}_1 \\ \alpha_i &= \boldsymbol{c}_i^T\boldsymbol{Q}_{i1}\boldsymbol{c}_i - 2\boldsymbol{x}_{id}^T\boldsymbol{Q}_{i1}\boldsymbol{c}_i + \boldsymbol{x}_{id}^T\boldsymbol{Q}_{i1}\boldsymbol{x}_{id} \end{aligned} \tag{19}$$

Since the two matrices $\boldsymbol{Q}_{i1}$ and $\boldsymbol{Q}_{i2}$ are both symmetric and positive defined, also the matrix Γ_i is positive defined and the objective function $J_i(\boldsymbol{r}_i)$ as given in (18) is a strictly convex quadratic form. Therefore, the optimization problem for each single agent i again has the form(7) where the allocation now is a trajectory of allocations over K time instants taking into account the discrete-time dynamics of each subsystem. The previously derived distributed solution algorithm can be applied again with the only difference that the vector $\boldsymbol{p}$ represents a price trajectory.

The overall market-based dynamic resource allocation now again requires the assignment of one single control agent i to each of the I entities or subsystems that need resources and the definition of one coordinator agent. Each of the control agents has a model of the dynamic behavior of its assigned local subsystem, i.e. the control agent i knows the matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ and it is able to measure the current vector of state variables, i.e. $\boldsymbol{x}_i(0)$. The coordinator agent then starts in iteration step $\kappa = 0$ with the definition of a first price trajectory $\boldsymbol{p}(\kappa = 0)$ which now has the dimension of K-times the dimension of one single price vector in one single instant. The price trajectory is distributed to all control agents which calculate their demand trajectories $\boldsymbol{r}_i(\boldsymbol{p}(\kappa))$ with the knowledge of the goal trajectories $\boldsymbol{x}_{id}$ over the next K time steps. This is done by solving the respective minimization problems (7) with the objective function (18).

All demand trajectories are transmitted to the coordinator agent which compares overall supply and demand and then adjusts the price trajectory as explained in the previous section. Again this iteration procedure stops until the overall (and also future market) is cleared and overall supply trajectory equals the overall demand trajectory. Then the allocation for that current time step is realized and the overall procedure starts again in the next time step. That means that the current allocation is calculated on the basis of model-based predictions of the future states, but the calculated future resource trajectory is not completely allocated. The reason for that approach is the possibility to consider disturbances of the state variables that can occur in the next time step. Thus the overall allocation scheme is similar to a model-predictive control algorithm but realized by distributed communicating control agents.

A second possible approach takes into account that the main goal of the local control agents is the control of the single subsystems via a suitable allocation of the resources. In a heating system for instance, each control agent has to allocate that amount of heating energy to the assigned room that a required temperature setpoint is maintained. This is done by a feedback of the room temperature and a conventional controller like a PID controller. There are many possibilities to develop suitable control algorithms and this is a well-known standard procedure in control engineering, see the corresponding literature. Without any loss of generality we can assume here that discrete-time state variable feedback controllers with a controller matrix $\boldsymbol{K}_i$ exist for all single subsystems $i = 1, \ldots, I$:

$$\tilde{\boldsymbol{r}}_i(k) = -\boldsymbol{K}_i \cdot \boldsymbol{x}_i(k) \tag{20}$$

Herein, $\tilde{\boldsymbol{r}}_i(k)$ is the allocation vector that is proposed from the pure control algorithm in each instant k. However, this computation only takes the fulfillment of the given control goals into account and not the fact that the overall allocated amount of the resources is limited to $\boldsymbol{r}$. For that purpose, each control agent also acts as a local optimizer in addition to the pure control task for the overall coordination of the resource allocation process. However, a suitable objective function $J_i(\boldsymbol{r}_i(k)$ has to be derived that again makes the market-based allocation scheme applicable.

Here, the following idea has been developed. If $\tilde{\boldsymbol{r}}_i(k)$ is the amount of the resources proposed for an allocation by the pure control algorithm in order to fulfill the local control goal and $\boldsymbol{r}_i(k)$ is the finally allocated amount negotiated by the control agent, the difference between these two vectors should be minimized, i.e.

$$\begin{aligned} J_i(\boldsymbol{r}_i(k)) &= (\boldsymbol{r}_i(k)-\tilde{\boldsymbol{r}}_i(k))^T \cdot (\boldsymbol{r}_i(k)-\tilde{\boldsymbol{r}}_i(k)) \\ &= \boldsymbol{r}_i^T(k)\boldsymbol{r}_i(k) - 2\cdot \boldsymbol{r}_i^T(k)\tilde{\boldsymbol{r}}_i(k) + \tilde{\boldsymbol{r}}_i^T(k)\tilde{\boldsymbol{r}}_i(k) \end{aligned} \quad (21)$$

This objective function $J_i(\boldsymbol{r}_i(k))$ again is a strictly convex quadratic form and therefore the proposed market-based allocation algorithm can be applied in each instant k. However, this objective function also takes the control goal into account, because it also contains the corresponding vector $\tilde{\boldsymbol{r}}_i(k)$.

The overall market-based dynamic resource allocation then works as follows. In each discrete time step k, each local control agent i measures the full vector $\boldsymbol{x}_i(k)$ of state variables of the associated subsystem. With the help of a local control algorithm, each control agent proposes a local allocation $\tilde{\boldsymbol{r}}_i(k)$ which is then given to an included local optimizer i. Each optimizer i then calculates the current objective function $J_i(\boldsymbol{r}_i(k))$. From the viewpoint of the optimization, this objective function is a pure static function at each instant k and the dynamic behavior of the subsystem i has been taken into account during the development of the state variable feedback controller. With the help of that separation between control task and resource allocation, a realization of the proposed system is not very difficult since the control algorithms itself remain unchanged. With the help of a coordinator agent and the previously described iterative algorithm, the actual allocations $\boldsymbol{r}_i(k)$ are then calculated. The optimizers within the control agents then command these values to their local controllers to realize this allocation with the suitable actuators. The overall physical structure of the system is shown in figure 4.

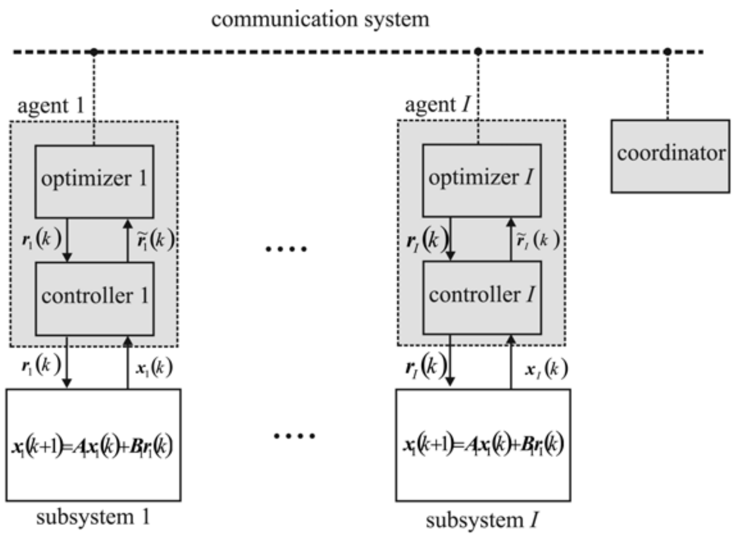

Fig. 4. Structure of the dynamic resource allocation which combines control and allocation task.

4. Application examples

In a first application example, we consider flow control in a network as a static distributed resource allocation problem given by (1) for a number of agents. The network is described by a directed graph with M nodes and N links between the nodes as given in Fig. 5. The nodes have no storage capability and act as routing elements. Each link L_j, $j = 1, \dots, N$ has a certain maximum transport capacity c_j and associated costs w_j for the transport over L_j. We assume several simultaneous transport requests where each comprises the injection of an input flow at a start node, the distributed transport over the inner links of the network and the extraction of the flow at one or several end nodes. Several of these requests with the related flows now have to be routed simultaneously through the network where the transport capacity of the single links are the resources that have to be shared and allocated to the requests. The goal of a single request is to minimize the costs of the overall transport. Such flow control problems without fixed single paths occur in packet-switched communication networks, in traffic systems and in energy and water supply networks.

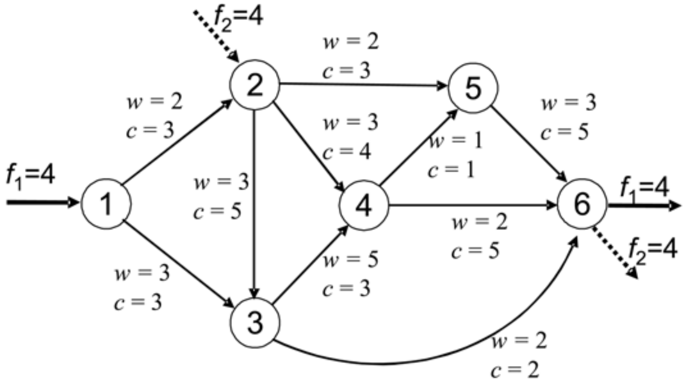

Fig. 5. Directed graph of the transport network.

Now each request is associated with a control agent i which is responsible for the allocation of the flows over each of the N links from the start to the end nodes. After the distributed allocation process between the agents is finished, this information is transmitted to the nodes for local routing. Each control agent has to determine the vector $\boldsymbol{r}_i \in \mathbb{R}_+^L$ of its corresponding allocations, where the single vector elements are ordered according to the link numbering, i.e $\boldsymbol{r}_{ij}$ is the amount of flow of request i through link j. The topology of the network given by the directed graph is described by its incidence matrix $\boldsymbol{C}$ that has M rows and N columns and the single matrix elements C_{kj} are

$$C_{kj} = \begin{cases} -1 & \text{if node } k \text{ is start node of link } j \\ 1 & \text{if node } k \text{ is end node of link } j \\ 0 & \text{else} \end{cases} \tag{22}$$

The matrix-vector product $\boldsymbol{C} \cdot \boldsymbol{r}_i$ now defines the divergence of the nodes which allows the formulation of a condition of the injection and extraction of the flows of request i at the respective nodes in the form

$$\boldsymbol{C} \cdot \boldsymbol{r}_i = \boldsymbol{d}_i \tag{23}$$

where the elements of the vector $\boldsymbol{d}_i \in \mathbb{R}^M$ are

$$d_{ik} = \begin{cases} < 0 & \text{injection of flow at node } k \\ > 0 & \text{extraction of flow at node } k \\ 0 & \text{else} \end{cases} \tag{24}$$

The vector $\boldsymbol{d}_1$ for the first transport request in the example given in Fig. 5 is

$$\boldsymbol{d}_1^T = (-4, 0, 0, 0, 0, 4)$$

which means an injection flow of magnitude 4 at node one and and extraction of that flow at node six. Now the single optimization problems for the control agents i can be formulated in the form (7) where the objective function $J_i(\boldsymbol{r}_i)$ considers the transportation costs, e.g. in a linear form $J_i = \boldsymbol{w}^T \cdot \boldsymbol{r}_i$ where $\boldsymbol{w}$ is the vector of all costs of the links. However, these objective functions could be any suitable strictly convex functions which only influences the local optimization algorithm for the single control agent i. With the help of the proposed interaction scheme the price vector $\boldsymbol{p}$ is adjusted in a way that the overall condition for the allocation holds

$$\sum_{i=1}^{I} \boldsymbol{r}_i \leq \boldsymbol{c} \tag{25}$$

where the vector $\boldsymbol{c}$ is the vector of all link transport capacities. This inequality condition then can be transformed into the equality condition of form (1) by the introduction of slack variables. The allocation performed by the market-based multiagent system results in

$$\begin{aligned} \boldsymbol{r}_1^T &= (2, 2, 0, 1.02, 0.98, 0, 2, 0, 1.02, 0.98) \\ \boldsymbol{r}_2^T &= (0, 0, 0, 2.4, 1.6, 0, 0, 0, 2.4, 1.6) \end{aligned}$$

An example considering resource allocation in a dynamic system is the distribution of heating energy in an office building. Here we consider the building as a number of room modules connected according to the topology of the building. Each room module i comprises four connected elements for thermal storage: the air, the floor, the outer wall (to environment) and the inner wall with the temperatures T_i, T_{Fi}, T_{WOi} and T_{WIi}. The connections and heat flows between the elements are shown in Fig. 6. In the whole model, we neglect radiation heat and only consider convection of heat. To each office, i.e. to the air of the office, a certain heat flow r_i can be allocated as a part of the overall heating energy resource. The offices have windows that allow some direct loss heat flow j_{Li} from the room to the environment. From the air, there is a heat flow j_{Fi} to the floor and heat flows j_{WOi} to the outer wall and j_{WIi} to the inner wall. Since the inner walls connect two neighboring offices, there is an additional heat flow j_{ik} from that wall to the other office k with air temperature T_k (or several other offices). The outdoor temperature T_{Ei} is modelled as a virtual variable which is a superposition of two parts. One part is the general outdoor temperature that is independent from the orientation to the sun and only depends on the time of the day. The second part describes the influence of the radiation of the sun and therefore depends on the orientation to the sun. Therefore there are different virtual outdoor temperatures for different offices in the simulation model.

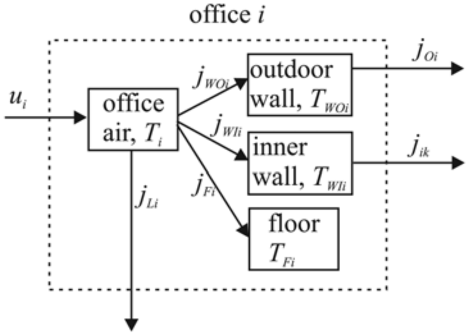

Fig. 6. Dynamic model of a single office.

The differential equation describing the air temperature T_i in office i under the assumption that the mass m_i of the air is constant is

$$c_i \cdot m_i \cdot \dot{T}_i = r_i - j_{Li} - j_{Fi} - j_{WIi} - j_{WOi} \tag{26}$$

with $j_{Li} = k_{Li} \cdot (T_i - T_{Ei})$, $j_{Fi} = k_{Fi} \cdot (T_i - T_{Fi})$, $j_{WIi} = k_{WIi} \cdot (T_i - T_{WIi})$ and $j_{WOi} = k_{WOi} \cdot (T_i - T_{WOi})$. Herein, c_i is the specific heat capacity of the air while the constants k_{Li}, k_{Fi}, k_{WIi} and k_{WOi} are the coefficients of heat transfer normalized to the effective surfaces, respectively. The other three elements for thermal storage floor, inner and outer wall are modelled in a similar way where the details are omitted here. One single room module i hence can be described by a state variable model of order four of the form

$$\begin{pmatrix} \dot{T}_i \\ \dot{T}_{Fi} \\ \dot{T}_{WIi} \\ \dot{T}_{WOi} \end{pmatrix} = \boldsymbol{A}_i \cdot \begin{pmatrix} T_i \\ T_{Fi} \\ T_{WIi} \\ T_{WOi} \end{pmatrix} + \boldsymbol{b}_i \cdot r_i + \boldsymbol{D}_i \cdot \begin{pmatrix} T_{Ei} \\ T_k \end{pmatrix} \tag{27}$$

where $\boldsymbol{A}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{D}_i$ are all constant matrices or vectors that depend on the masses, heat capacities and coefficients of heat transfer. The matrix $\boldsymbol{D}_i$ describes the influence of the disturbances, i.e. the outdoor temperature and the temperatures of the neighboring office(s). In general, the trajectory of the outdoor temperature can be estimated and predicted while the trajectory of the temperature of the neighboring offices are unknown since they are a result of the allocation algorithm. However, if the heat transfer through the wall is not too big, these interconnections can be neglected with respect to the disturbance caused by the outdoor temperature. From a mathematical point of view however, the consideration of the predicted trajectory of the outdoor temperature only leads to an additional part in the vector $\tilde{\boldsymbol{a}}_i$ in (14).

Each office is associated with a control agent i that is responsible for the resource allocation, i.e. the allocation of heating energy r_i. The state variable model (27) can be transformed into a discrete-time state variable model of form (12) for each single office i. The goal of the resource allocation to office i is to minimize the difference between the trajectory of the air temperature (which can only be controlled by the input r_i) and a desired temperature trajectory while the energy consumption should be minimized simultaneously over an interval of K time steps. This leads to a definition of the objective function J_i as described in (17). Herein the matrix $\boldsymbol{Q}_{i1}$ has only the first element on the main diagonal as a non-zero element since there are no desired trajectories for the other temperatures of the walls and floor.

The dynamic allocation then can be processed by the agents as described in the previous section. Each control agent has the knowledge of the local model (27) while neglecting the interconnections between the offices and using an estimation of the future outdoor temperature. The control agent calculates the allocation trajectory for the next K time steps and all trajectories are balanced by the distributed coordination with the price trajectory. Then all control agents realize their local allocation for the next time step only and the overall procedure starts again. Especially the prediction of the future outdoor temperatures as the main disturbances leads to an early adaptation of the single allocations taking into account the inert behavior of a heating process. Thus the overall efficiency of the heating system and the comfort is considerably increased. One result for a building configuration with twelve offices is shown for four rooms in Fig. 7. Here it becomes obvious that the set-point temperatures are maintained except in intervals where the radiation of the sun is too strong (see offices four and eleven). These deviations are caused by the fact that in spite of zero allocation of heating energy the incoming heat still leads to an increase of the temperature. However, this problem could only be solved by a cooling system. Nevertheless, the algorithm works well in the simulation and the overall heating energy is always kept below a desired maximum.

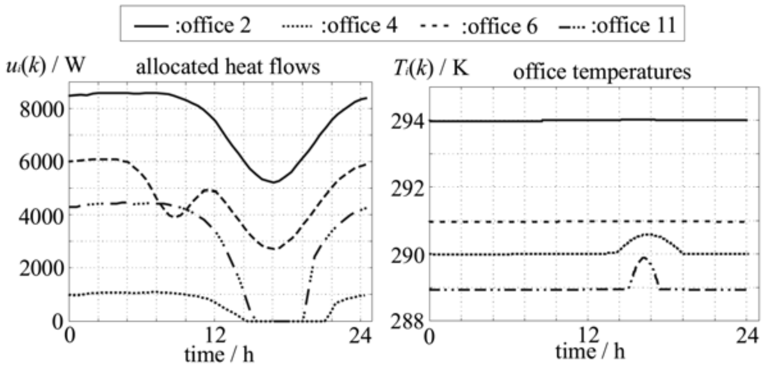

Fig. 7. Results of heat allocation for a simulated building.

The second proposed agent-based dynamic resource allocation scheme is finally applied to a real industrial production process in sugar industry. In Europe, sugar is mainly produced by extracting it from sugar beets. This extraction is performed by cutting the beets into small parts and giving them into hot water. Within that heating process, the sugar is dissolved in the water and steam is generated. In a following production step, the sugar cristallization, the sugar-water solution is given into several cooking stations where this solution is further heated until the solution is oversaturated and the sugar starts to cristallize. Hereby, the steam that is generated in the sugar extraction is then used in this second process of cristallization as heating energy, see Fig. 8. The overall amount of steam should be kept constant in order to avoid disturbances during the extraction. Therefore, the limited and constant amount of the resource “steam” has to be allocated in a suitable way to the different cooking stations and that resource allocation is accomplished here with the proposed distributed market-based algorithm using agents.

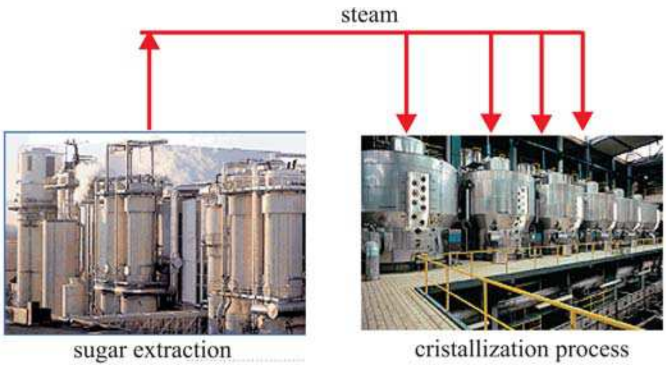

Fig. 8. Steam allocation in the sugar production process.

The resource allocation problem can be expressed as follows. Given the overall constant amount j_H of steam generated during the extraction, this steam must be allocated to a number of I cooking stations, each indexed by i. At each discrete time step k, the amount of steam allocated to a single cooking station i is $j_{H,i}(k)$ and the allocation problem expressed in the form (1) yields

$$\begin{aligned} &\min_{\{j_{H,i}(k),\forall i\}} \sum_{i=1}^{I} J_i(j_{H,i}(k)) \\ \text{s.t.} \quad & j_{H,i}(k) \geq 0 \quad \forall i \\ & \sum_{i=1}^{I} j_{H,i}(k) = j_H \end{aligned} \tag{28}$$

However, the allocation of the limited overall amount of steam at instant k to the different cooking stations must be done in a suitable way and the allocated amounts of steam depend on the current state of the stations and the control goal. A single cooking station is shown in Fig. 9.

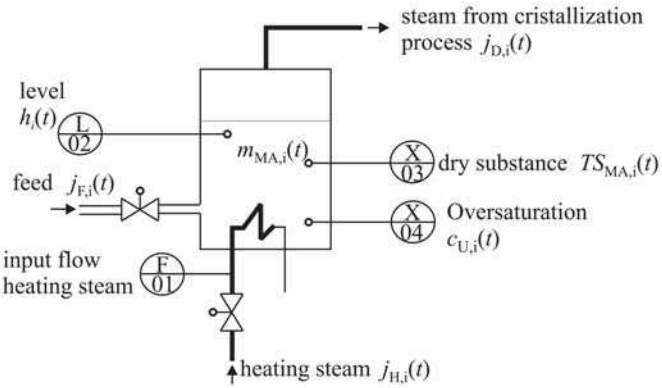

Fig. 9. A simplified scheme of a cooking station.

Herein, $m_{MA,i}$ is the overall mass, $j_{F,i}$ is the amount of sugar-water solution that flows into the tank, $j_{H,i}$ is the heating steam, $C_{U,i}$ is the oversaturation, $TS_{MA,i}$ is the dry substance and h_i is the level in the tank. Concerning that process, a discrete-time state space model was developed [Voos, 2003] in order to design a local state variable feedback controller as given in (20). Here, the main control goal is to keep the oversaturation constant while filling the tank with the solution and to reach a dry substance content in the solution of 90% at the end of the batch process. The input variable for that control task is the amount of the actually allocated amount of heating steam $j_{H,i}(k)$. The allocation proposed by this local controller then can be expressed here as

$$\tilde{j}_{H,i}(k) = -\boldsymbol{K}_i \cdot \boldsymbol{x}_i(k) \tag{29}$$

That leads to the formulation of the local objective function corresponding to (21) and the proposed distributed allocation algorithm can be applied.
The approach is currently tested in a simulation where some results are given in figure 10. Here a number of 3 cooking station is supplied with an overall amount of j_H = 8 kg/sec of heating steam. The different cooking stations are started at different start times and therefore the batches also end at different stop times. In figure 10, the three different allocated amounts $j_{H,i}(k)$ as well as the sum of all of them are depicted. It is obvious that the algorithm succeeds in keeping the sum constant and equal to the required overall amount j_H = 8 kg/sec of generated steam. In addition, in each single batch the control goals are fulfilled

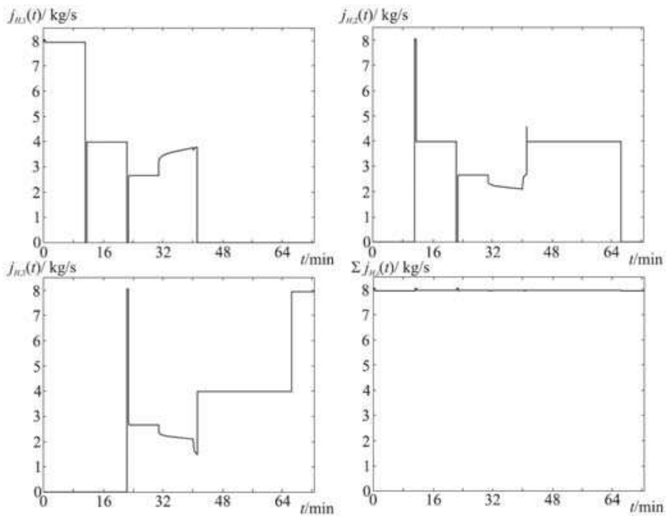

Fig. 10. Allocated amounts of steam in the simulation.

(not shown here). This example also illustrates the flexibility of the market-based approach since it can handle a flexible and varying number of agents (i.e. batches here) that take part in the allocation. At the beginning, only one agent takes part in the allocation process since only the first batch has been started. Then with the start of the second batch process, a second agent starts to take part in the allocation and so on.

5. Conclusions and future works

In this chapter a new approach for agent-based distributed resource allocation has been derived which is especially suited to cope with allocations in dynamic environments. The interaction scheme is inspired by economic markets and therefore called market-based allocation. Herein, the resource allocation problem is formulated as an optimization problem that can be decomposed into single optimization problems that only depend on a global price vector. That price vector is used to balance the overall allocations, i.e. the solutions of the single optimization problems. A general market-based approach has been extended here in two directions. First, more general objective functions now can be used which leads to a wider range of applications. Second, the idea of allocation to dynamic systems has been included. The approach is demonstrated using three technical examples for proof of concept. One interesting question that has to be solved in the future is the consideration of allocation to dynamic systems where a dynamic interconnection between the subsystems occurs.

6. References

[Unland, 2003] R. Unland, M. Klusch, M. Calisti, (Ed.) *Software Agent-Based Applications, Platforms and Developments Kits*, Birkh¨auser, Basel 2005.

[Maturana, 2006] F. Maturana et. al., *Agent Infrastructure for Distributed Control and Interoperability.* In proc. of the IEEE Workshop on Distributed Intelligent Systems DIS 2006, Prague, Czech Republic, 2006.

[Jennings, 1998] N.R. Jennings, K. Sycara, M.Wooldridge, "A Roadmap of Agent Research and Development", *Autonomous Agents and Multi-Agent Systems*, vol. 1 (1), pp. 7-38, 1998.

[Clearwater, 1996] S.H. Clearwater (ed.), *Market-Based Control: A Paradigm For Distributed Resource Allocation.* Singapore: World Scientific, 1996.

[Debreu, 1959] G. Debreu, *Theory of value.* New Haven and London: Yale University Press, 1959.

[Ibaraki, 1988] T. Ibaraki and N. Katoh, *Resource Allocation Problems - Algorithmic Approaches.* Cambridge (MA): MIT Press, 1988.

[Voos, 2003] H. Voos, *Market-based Control: Eine neue Methode zur automatisierten Ressourcenzuteilung.* (in German.) Ph.D. Thesis, Aachen: Shaker, 2003.

[Voos, 2006] H. Voos, *Agent-Based Distributed Resource Allocation in Technical Dynamic Systems.* in Proc. of the IEEE Workshop on Distributed Intelligent Systems DIS 2006, Prague, Czech Republic, 2006.

[Voos, 2007] H. Voos, *Resource allocation in Continuous Production using Market-Based Multiagent Systems.* in Proc. of the IEEE 5th International Conference on Industrial Informatics INDIN 2007, Wien, Austria, 2007.

[Colombo, 2004] A.W. Colombo, R. Schoop, R. Neubert, *Collaborative (Agent-based) Factory Automation*. Chapter 109 in "The Industrial Information Technology Handbook.", Richard Zurawski (Ed.), Boca Raton, USA,: CRC Press LLC, 2004.

[Sandholm, 1999] T. W. Sandholm, "Distributed Rational Decision Making", in Multiagent Systems, G. Weiss, ed., pp. 201-258, Cambridge: MIT Press, 1999.

[Stoer, 1993] Stoer, J.: *Numerische Mathematik 1.* 6. Aufl. Berlin, Heidelberg: Springer Verlag 1993

[Weiss, 1999] G. Weiss, ed.; *Multiagent Systems.* Cambridge, MA: MIT Press, 1999.

[Bussmann, 2003] N. Jennings, S. Bussmann, *Agent-Based Control Systems.* In IEEE Control Systems Magazine, Vol. 23, June 2003.

[Kluegl, 2005] F. Klügl, A. Bazzan, S. Ossowski (Eds.), *Applications of Agent Technology in Traffic and Transportation.* Basel: Birkhäuser 2005.

[AAMAS, 2008] Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems AAMAS 2008, Lissabon, Portugal, 2008.

[Cervenka, 2007] R. Cervenka, I. Trescansky, *The Agent Modeling Language - AML.* Berlin, Heidelberg: Springer 2007.