Task Execution in Distributed Smart Systems

Uwe Jänen[1(✉)], Carsten Grenz[1], Sarah Edenhofer[1], Anthony Stein[1], Jürgen Brehm[2], and Jörg Hähner[1]

[1] Lehrstuhl für Organic Computing, Institut für Informatik, Universität Augsburg, 86135 Augsburg, Germany
uwe.jaenen@informatik.uni-augsburg.de
[2] Leibniz Universität Hannover, Institut für Systems Engineering, Fachgebiet System- und Rechnerarchitektur, 30167 Hannover, Germany

Abstract. This paper presents a holistic approach to execute tasks in distributed smart systems. This is shown by the example of monitoring tasks in smart camera networks. The proposed approach is general and thus not limited to a specific scenario. A job-resource model is introduced to describe the smart system and the tasks, with as much order as necessary and as few rules as possible. Based on that model, a local algorithm is presented, which is developed to achieve optimization transparency. This means that the optimization on system-wide criteria will not be visible to the participants. To a task, the system-wide optimization is a virtual local single-step optimization. The algorithm is based on proactive quotation broadcasting to the local neighborhood. Additionally, it allows the parallel execution of tasks on resources and includes the optimization of multiple-task-to-resource assignments.

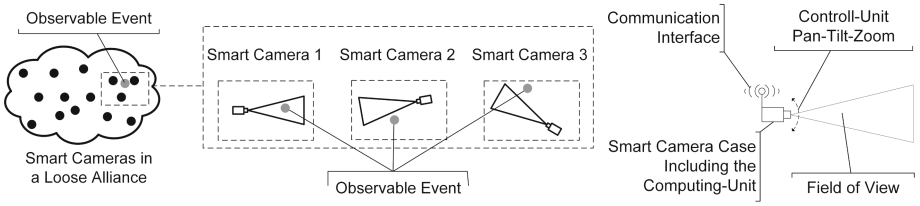
Keywords: Job-resource-model · Optimization transparency · Virtual local single-step optimization · Proactive quotation-based optimization · Multiple-task-to-resource assignment

1 Introduction

The scientific achievements of the past years enforced a rapid augmentation of computerized systems, so called *smart systems*. From the perspective of the authors, this smart trend is driven by standardization and modularization of system components, the increase of computing power and capabilities of individual system components and their interconnection. This trend makes new applications possible. The configuration space of individual systems and their possibilities for collaboration have increased significantly. Former passive systems now can actively service novel application scenarios. Example scenarios are smart home, smart desktop-grids and distributed smart camera networks. In a smart home a possible application may be to illuminate the kitchen by a defined luminous flux. Therefore, a smart electric stove with a lamp, a smart table lamp or a smart ceiling lamp can be used to fulfill this task. Another example is a smart desktop-grid with the task of executing a simulation. Therefore, different

G. Di Fatta et al. (Eds.): IDCS 2015, LNCS 9258, pp. 103–117, 2015.
DOI: 10.1007/978-3-319-23237-9_10

desktop configurations are available, like multi-core CPU and GPU. Within this work we focus on networks with pan-tilt-zoom capable smart cameras, because it is one of the most challenging scenarios. In a smart camera network, as illustrated in Fig. 1(a), the task could be to observe an event with different cameras. Figure 1(b) shows the system architecture of a smart camera. The new possibilities to fulfill a task, increase the configuration space rapidly. So, one question comes up: Which task has to be fulfilled by which smart system component? The first research subject is to create a model to handle this new possibility to fulfill tasks by different smart system components. The second research subject is the optimization of task-to-smart-system-component assignment in naturally distributed systems with only local algorithms. Both are explained in the following Sects. 1.1 and 1.2. In the remainder of this paper, we will present an approach to handle the first and second research subject. Therefore, a job-resource model will be introduced in Sect. 2. Based on this model, a heuristic approach for virtual local single-step optimization is presented in Sect. 3. This approach is capable for parallel execution of tasks on a smart component within a distributed system. The algorithm will be evaluated on optimization speed and the capability to solve generalized assignment problem benchmarks in Sect. 6.



(a) Observation in a smart camera network. (b) Smart Camera

Fig. 1. Distributed smart system exemplified as distributed smart camera network.

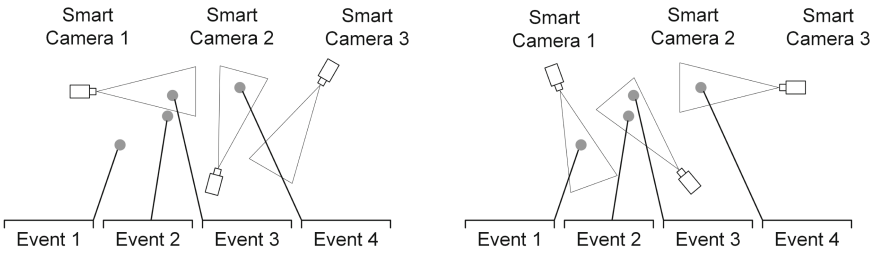
1.1 First Research Subject: Model Creation

In smart systems, different tools are available to satisfy different tasks. The model must be independent from the task, the optimization criterion and the smart system component. Generally, we have to decide whether we prefer centralized, local or hybrid algorithms in a natural distributed system. On one hand, this model must support user-objectives such as turn on the light in the kitchen or rather the observation of an event in a smart camera network. On the other hand, the system has to support system-wide objectives as well as the optimization to the resulting global criterion.

1.2 Second Research Subject: Optimization Transparency

In a distributed system using only local knowledge and local algorithms, it is hard to optimize on global criterion. This is illustrated in Fig. 2. Here, the scenario consists of three smart cameras (SC) and four events to observe. Initially

in Fig. 2(a), only event 3 and event 4 are observed. The system objective is to observe as many events as possible. In Fig. 2(b) the obviously best alignment of the cameras field of view (FoV) is shown. The challenge is to optimize the configuration with local knowledge and local algorithms from the situation depicted in Fig. 2(a) as close as possible to the optimal state in Fig. 2(b). The system-wide optimization shall be transparent to the system. This means it will not be visible to the participants. In particular, we enforce a virtual, local single-step optimization (VLSO).



(a) Initial situation with three smart cameras and four observable events.

(b) Obviously best smart camera alignment to cover all events.

Fig. 2. Example for an initial situation within a distributed smart camera network. The network consists of three smart cameras and four events to observe.

2 Job-Ressource-Model to Handle the First Research Subject

To achieve universality and independence of explicit application scenarios, the model should contain as much order as necessary and as few rules as possible. A smart system consists of smart components. Based on the smart camera definition in [1], a smart component consists of a computing- and a communication-unit. In addition, it can be optionally extended e.g. by sensors, actuators etc. (Fig. 1(b)). A smart component can also consist of multiple other smart components. Each smart component has a *neighborhood RN* which is connected ad-hoc (e.g. wlan). Smart components have a *neighborhood relation* if they can communicate directly with each other. Two smart cameras are in a neighborhood if they are in the same broadcast domain (layer-2 broadcast in the OSI-model). Messages will not be forwarded beyond this neighborhood (routing). A smart system is used to fulfill different objectives, so called *micro-* and *macro-objectives*. How are they related to the smart components? Micro-objectives are individual objectives which result in a software instance i.e. a program in the main storage of a smart component, which can be executed, and the associated data. Macro-Objectives are superior system-objectives. As an example, we consider a smart camera network in which two glass break sensors detect events (A and B). This will automatically create two observation tasks. These micro-objectives are

given to the smart camera network and are represented by software instances. Both instances compete for the smart cameras. Which instance aquires which camera has to be determined by the interaction of the software instances. This interaction is defined by the macro-objective of the system. The macro-objective could be the maximization of a system-wide observation success for example. We developed a *job-resource-model* based on this *objective-model*. A smart system component i will further be interpreted as a resource r_i from a set of resources R. A software instance, which pursues a micro-objective n and is bounded by the interaction rules of macro-objectives, is further called job a_n, an element of the set of all jobs A.

2.1 Resources

A resource r_i is the reduced representation of a smart component, which can be allocated by a job a_n to fulfill its micro-objective. A resource is defined by its ID i and the estimated success it will provide to a job's micro-objective running on it. The success of job a_n on resource r_i is denoted by P_i^n. If a resource i knows about a micro-objective, it broadcasts information about it proactively to its neighborhood RN_i. This proactive information broadcast is also called the resource-to-resource-interaction (R2R-interaction). A non-existing resource (neutral element) will be denoted by $r_\varnothing$.

2.2 Jobs

A job a_n is a representation of an individual objective, respectively a software instance. This software instance consists of a management part and an execution part. The management part is responsible for a self-organized allocation of a resource. The execution part is e.g. a thread for person detection in single images. A job *is on* a resource, if a software-instance is located on the corresponding smart component. A job *allocates* a resource, if the execution part of the software instance is running on the smart component. A job, which is on a resource, has full access to the data on that resource. Jobs can interact by *sending messages* and allocating a resource by *displacing an other job*. This job-to-job-interaction (J2J-interaction) is defined by the rules of the macro-objectives. The non-existing job, respectively the idle job on a resource, is denoted by $a_\varnothing$. The idle job is the only job which can allocate an idle resource and will do this automatically. The job-to-resource-interaction (or J2R-interaction) then is defined as follows: a job can allocate a resource and exchanges information, if the job is on the resource. After defining the job-resource-model we can focus on the macro-objectives which are congruent to the second research subject in the following section.

3 Proactive Quotation-Based Scheduling to Handle the Second Research Subject

In the previous section, we introduced an objective-model and derived a job-resource-model from it. The objective-model distinguishes between micro- and

macro-objectives. The micro-objectives are individual objectives and macro-objectives are objectives regarding the whole system. The only admitted way to enforce macro-objectives in the presented job-resource-model, is to affect the J2J-interaction. That means the information which has to be exchanged between jobs and the rules, when a job is allowed to displace another job. So, we need a local algorithm which is capable to optimize system wide. Such an algorithm shall avoid long negotiation chains within the whole system. To enlight the challenge, we present a scene with three smart cameras in Fig. 3. SC1 and SC2 are observing the events 1 and 2. SC3 is idle. At time stamp T' a glass break sensor detects an additional event 3. The micro-objective to observe this event by a smart camera is represented by a job. On the right of Fig. 3, the related gantt charts to each resource and each job are depicted. Each bar represents the duration for how long the event can be observed. The predicted success P_i^n for executing job a_n on resource r_i is denoted at the end of each bar. The macro-objective is to increase the system wide observation success:

$$P_{sys} = \sum_{a_n} P_i^n \cdot [a_n \text{ allocates } r_i] \tag{1}$$

with Iverson brackets as $[statement] = 1$ if $statement == true$ and $[statement] = 0$ if $statement == false$. It is obviously the best solution that SC1 adjusts its FoV to observe event 3 at time stamp T'. Then, SC2 switches its FoV on event 1 and, additionally, SC3 turns on event 2. The challenge is to avoid long negotiation chains such as job 3 asks job 1 to handover the resource SC1. Then, job 1 has to ask its neighborhood to change on another smart camera and so on. In general, this simple approach will result in a tree search. As already said, we try to achieve a transparent optimization with local algorithms in distributed smart systems. In particular, we want to achieve a virtual, local single-step optimization (VLSO): From a job's point of view, it shall seem to be a local optimization, which only needs one step to complete. Such an algorithm can be found in [3] and it is called *Proactive Quotation-Based Scheduling (PQB)*. This algorithm will be shortly introduced in the following subsection. Afterwards, we will extend the PQB algorithm to handle a parallel execution of several micro-objectives on single resources.

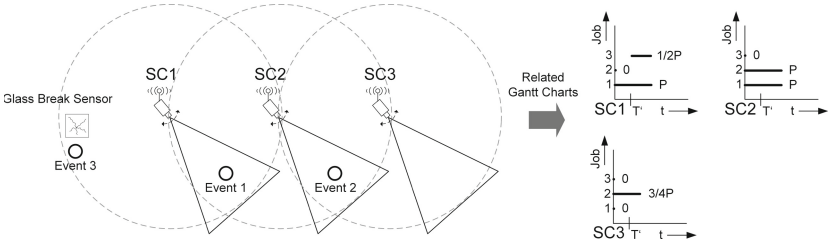


Fig. 3. Network with three smart cameras and two events. At time stamp T' an additional event occurs.

3.1 Single Micro-Objective to Single Resource Association

The main idea is: Each job a_n allocating a resource r_i locally broadcasts a *quotation* Q_i^n, which has to be fulfilled by another job to displace the offering job. This is illustrated in Fig. 4, following the example in Fig. 3. The resources proactively share the possible success of each job they know about with the neighborhood (Fig. 4(a)). One has to consider that this information will not be forwarded. Additionally, each job allocating a resource will send its quotation (Fig. 4(b)). At time stamp T' job a_3 is created. The resources again proactively broadcast the possible success (Fig. 4(c)). Job a_3 knows about the conditions to displace job a_1 because of the former quotation exchange (Fig. 4(b)). Then, job a_1 will displace job a_2. In the last step job a_2 will displace the idle job on resource r_3 (Fig. 4(d)). This illustration leads to the question: What was that quotation which causes job a_3 to displace job a_1 and temporarily decrease the local and system-wide observation success? The PQB-Scheduling algorithm is based on a *local mapping* of possible system-wide success improvement on a virtual local improvement of each job. In the following, a quotation Q_i^n is called *sales quotation*, because a virtual price has to be paid by a job a_x to receive the resource r_i from job a_n. In this context, success is equivalent to virtual money. A sales quotation solely implies that, if job a_n releases its resource r_i, the beneficiary job a_x has to compensate the loss:

$$Q_i^n = P_i^n \tag{2}$$

If job a_n can migrate to an alternative resource $r_{i'}$, the quotation has to be reduced by the predicted success on that resource. To create a low priced quotation for the resource, the job avoids to the resource with the maximum success:

$$Q_i^n = P_i^n - \max_{i'}\{P_{i'}^n\} \tag{3}$$

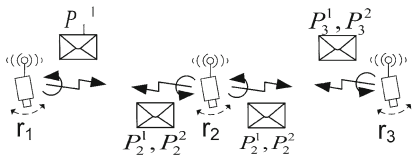
If job a_n has to ransom the alternative resource $r_{i'}$ from an offering job $a_{n'}$, the possible success on resource $r_{i'}$ has to be reduced by these costs. The quotation to be broadcasted by job a_n on resource r_i are given by:

$$Q_i^n = P_i^n - \max_{i'}\{P_{i'}^n - Q_{i'}^{n'}\} \tag{4}$$

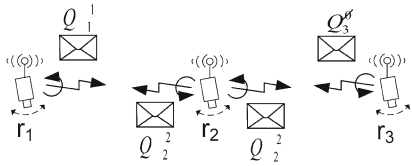
A job a_n that receives the quotation, has to decide if its current success is greater than the possible success on resource $r_{i'}$. The possible success is given by the success $P_{i'}^n$ subtracted by the costs $Q_{i'}^{n'}$.

$$P_{i',red}^n = P_{i'}^n - Q_{i'}^{n'} \tag{5}$$

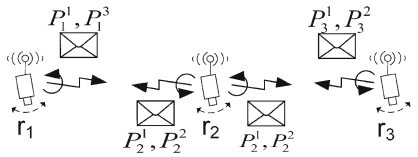
This approach by itself is not capable to achieve the solution we aimed for (see Fig. 2(b)), because grouping multiple micro-objectives on one resource is not supported. In the following, we extend the PQB-Scheduling algorithm to cope with this challenge.



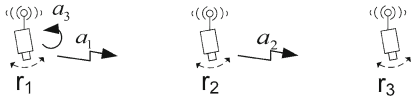
(a) Exchange of the possible job successes within the resource neighborhoods RN_i by proactively R2R-Interaction.



(b) Exchange of quotations within the resource neighborhoods RN_i by J2J-interaction.



(c) At time stamp T' job a_3 occurs. Job a_3 knows about the conditions to displace job a_1 because of the former quotation exchange.



(d) Job a_3 displaces a_1. Afterward job a_1 displaces a_2 and subsequently job a_2 displaces the idle job $a_\varnothing$ on r_3.

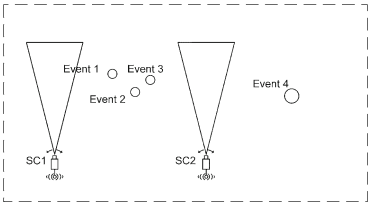
Fig. 4. Message exchange within the proactive quotation-based scheduling.

3.2 Multiple Micro-Objective to Single Resource Assignment

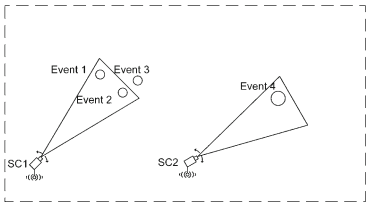
The calculation of a system-wide optimal grouping of micro-objectives is a hard problem. There are two general challenges and two problems to adapt the PQB-Scheduling algorithm to this scenario.

General Problem 1: The success of multiple micro-objectives executed on a single resource is the sum of the single micro-objective success within that group. The first problem is the dependency of a single micro-objective success on the group composition. Simply explained: If job a_1 and job a_2 share a resource r_1, usually $P_1^{a_1 \subseteq \{a_1, a_2\}} \neq P_1^{a_1}$. This is caused by the sharing of resources. In example Fig. 2(b), smart camera 2 is for instance not capable to adjust the focus on event 2 and 3 in the same quality as observing only event 2 or event 3.

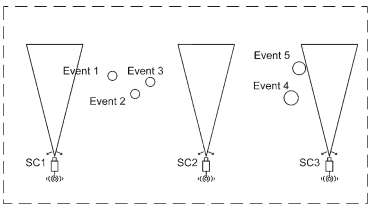
General Problem 2: The next challenge is to create the groups using only local knowledge and local algorithms. Two oppositional approaches are conceivable. In the first approach, the jobs form groups by themselves. The second approach is to build all possible combinations and only the micro-objective within a group with the highest success will be actively executed. At first sight, the dynamic grouping is preferable, because the amount of software-instance increases linearly with the number of micro-objectives. The amount of all possible combinations increases exponentially. Unfortunately, the dynamic grouping is hard with the local knowledge and local algorithms restrictions we choose. An example to show this issue is depicted in Fig. 5. In Fig. 5(a), the system view of SC1 is depicted. SC1 is not informed about SC3 and event 5 as shown in Fig. 5(c). So, with this different knowledge, the optimal camera alignment differs. With the knowledge



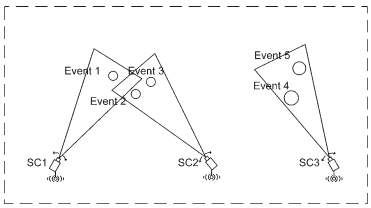
(a) System knowledge from the point of view of SC1.



(b) Best camera alignment from the point of view of SC1.



(c) System knowledge from the point of view of SC2.



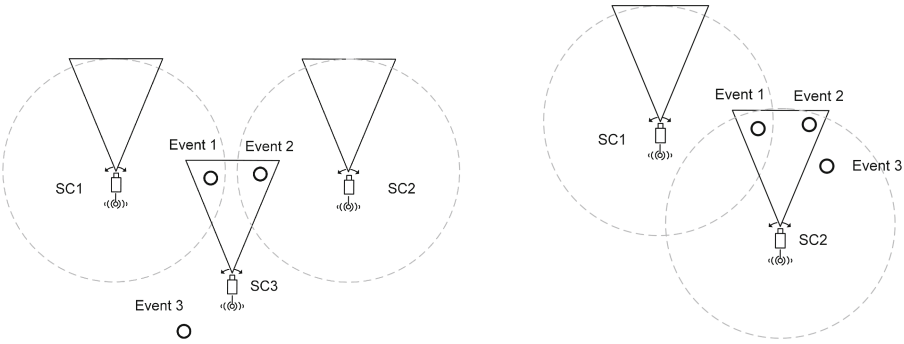
(d) Best camera alignment from the point of view of SC2.

Fig. 5. Dynamic grouping on runtime with local knowledge and local algorithms is not possible.

of SC1, the events 1 and 2 should be grouped. With the knowledge of SC2, event 2 and 3 as well as event 4 and 5 should be grouped.

PQB-Scheduling Problem 1: When calculating a quotation in the PQB-Scheduling approach, the splitting of a group has to be considered. In Fig. 6(a) an example is depicted. SC3 is observing event 1 and 2. SC1 and SC2 are not capable to observe both. The best solution will be that event 1 is observed by SC1, event 2 by SC2 and event 3 by SC3. So, the calculated quotation must consider the separation of event 1 and 2. Otherwise, the job that is responsible for observing event 3 is not able to buy SC3. This will increase the search space exponentially.

PQB-Scheduling Problem 2: When calculating a quotation in the PQB-Scheduling approach, the part of a group that remains on the resource has to be taken into account. Figure 6(b) shows an example. The SC2 is observing event 1 and 2. The best solution will be that event 1 is observed by SC1 and event 2 and 3 by SC2. So, the calculated quotation must consider the separation of event 1 and 2 and, additionally, that job a_1 observing event 2 stays on SC2. Otherwise, the job responsible for observing event 3 is not able to buy SC2. The received quotation has to be adopted by the receiving group. Unfortunately, this term cannot be calculated. The quotation is calculated under the constraint that no parts of the group remain on the resource. Theoretically an alternative more suitable quotation can exist.



(a) PQB-Scheduling Problem 1 (b) PQB-Scheduling Problem 2

Fig. 6. Example scenarios for PQB-Scheduling Problem 1 and 2

Formal Description of General Problem 1 and 2 And PQB-Scheduling Problem 1 and 2: Assuming a group g is allocating resource i. If group g leaves the resource r_i, the displacing job has to compensate the loss of group g. That equals to Eq. 2.

$$Q_i^g = P_i^g \tag{6}$$

If sub-groups g_{sub} of g are able to split up on several other resources $r_{i'}$, this success on the alternative resources $P_{i'}^{g_{sub}}$ has to be subtracted from Q_i^g.

$$Q_i^g = P_i^g - \max_{i', g_{sub}} \{\sum_{i'} (P_{i'}^{g_{sub}})\} \tag{7}$$

Here, the groups offering the alternative resources i' are donated with g'. It must be remarked that some parts g'_{sub} of the displaced groups g' can be integrated into the sub-groups g_{sub}.

$$Q_i^g = P_i^g - \max_{i', g_{sub}, g'_{sub}} \{\sum_{i'} (P_{i'}^{g_{sub} \subseteq \{g_{sub} \cup g'_{sub}\}})\} \tag{8}$$

The price $Q_{i'}^{g'}$ has to be paid to displace a group g' on resource i'. Again, the following has to be noted: if a sub-group of g' can be integrated, the price has also to be adopted to $Q_{i'}^{g' \setminus g'_{sub},\ g'_{sub} \cup g_{sub}}$.

$$Q_i^g = P_i^g - \max_{i', g_{sub}, g'_{sub}} \{\sum_{i'} (P_{i'}^{g_{sub} \subseteq \{g_{sub} \cup g'_{sub}\}} - Q_{i'}^{g' \setminus g'_{sub},\ g'_{sub} \cup g_{sub}})\} \tag{9}$$

Equation 9 states a hard combinatorial problem. Additionally, as mentioned in PQB-Scheduling Problem 2, the term $Q_{i'}^{g' \setminus g'_{sub},\ g'_{sub} \cup g_{sub}}$ can not be calculated exactly on the base of $Q_{i'}^{g'}$, but it can be estimated. On first sight, the problems seem to be solved with these equations. But they only represent the problems in a formal way. The term $\max_{i', g_{sub}, g'_{sub}}$ represents the computationally intensive part. For calculating a quotation, this part represents how to spread the micro-objectives on neighboring resources $\{i', ..., I'\}$.

All these equations contain groups g. But so far it is not clear where these groups came from.

4 Heuristic Approach for Parallel Execution of Multiple Micro-Objectives on Resources

We have developed a heuristic solution that solves all these problems in one stroke. The algorithm is based on a hybrid version of 'dynamic grouping of micro-objectives' and the 'parallel consideration of all combinations'. For clarity, throughout the explanations of the following heuristic approach, we will refer to the example depicted in Fig. 7. It shows a smart camera network initially consisting of three smart cameras (SC1 - SC3). As depicted, only SC1 is able to observe event 1-4 simultaneously. SC2 can only observe event 3 and 4, as well as SC3 can only observe event 1 and 2. Initially, job $a_{1,2,3,4}$ is executed on resource r_1. Resources r_2 and r_3 are idle. At time stamp T', event 5 occurs which can only be observed by SC1.

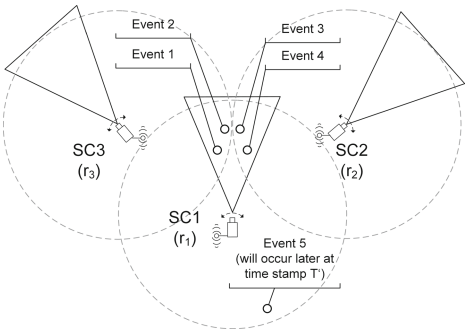


Fig. 7. Initial situation. Reference example for the heuristic approach.

First Step: On each resource, one job is created representing all micro-objectives which are executable on that resource: a_{all}. In the illustrated example, these will be $a_{1,2,3,4}$ for r_1, $a_{3,4}$ for r_2 and $a_{1,2}$ for r_3.

Second Step: Each job $a_{n,...,N}$ searches for an alternative resource (initially only a_{all}) and creates an *optimization matrix*. In the left table of Table 1, the optimization matrix of job $a_{1,2}$, which is on resource r_2, is depicted. The micro-objectives $m, ..., M$ are the ones currently allocating the resources in the neighborhood $r_i, ..., r_I$. This is denoted by an 'X' within the table. The row 'displaced' will be needed later.

Third Step: The micro-objectives $n, ..., N$ will be randomly assigned to the resources $r_i, ..., r_I$. The micro-objectives $m, ..., M$ can only remain on the resource or change to $r_\varnothing$. This assignment is also random. Unallowed assignments are marked with a '-' in the optimization matrix. If one of the micro-objectives $n, ..., N$ is already executed on a resource, the corresponding 'X' has to be deleted and it has to be marked in the line 'displaced'. Every row in the matrix with a micro-objective of $n, ..., N$ represents a possible group of micro-objectives on that resource. Only these rows will be considered for the calculation of the matrix

success. In the right table of Table 1, this is shown for job $a_{1,2}$ on resource r_2. This local solution will be rated. The success of that group and the costs to buy the resource has to be calculated: $P_{i'}^{g_{sub} \subseteq \{g_{sub} \cup g'_{sub}\}} - Q_{i'}^{g' \setminus g'_{sub},\ g'_{sub} \cup g_{sub}}$. In the example, this will be $P_3^{1,2} - Q_3^{a_\varnothing}$. For any displaced job, an extra charge of the success of that objective within the group on that resource has to be paid, $P_i^{a_n \subseteq g}$. In the example, this will be $P_1^{1 \subseteq \{1,2,3,4\}}$ and $P_1^{2 \subseteq \{1,2,3,4\}}$. The third step is repeated multiple times to find the matrix with the highest success. This exploration needs a heuristic. We used stochastic tunneling (ST) [10]. In the ST heuristic a change in the optimization-matrix will occur with a certain probability which depends on the current success compared to the former success of that matrix.

Fourth Step: This step starts, when the former step has terminated. If the success of the matrix calculated in the former step is greater than 0, the following will happen: Each row represents an assignment of micro-objectives to a resource. If the micro-objectives $n, ..., N$ are assigned to a single resource $r_{i'}$, (not including any element of $\{m, ..., M\}$ which is not also included in $\{n, ..., N\}$) this job will displace the offering job on that resource $r_{i'}$. If the micro-objectives $n, ..., N$ are associated to different resources, corresponding to each of these associations, a time-limited job is created, which starts in the second step of this algorithm. Time-limited means that the job will remove itself from the system when a timer expires. The time-limit will be reset after allocating a resource and will not be decreased until it is displaced. In the example, this means, job $a_{1,2}$ may displace $a_\varnothing$ on resource r_3, if the success of that matrix is greater than 0. Keep in mind, the extra charge (see displace row) may be higher than the success, so job $a_{1,2}$ will not displace $a_\varnothing$ on resource r_3.

An additional advantage of this heuristic is, it also can be used to calculate a quotation. Only the listed resources do not include the resource allocated by the job itself. Below on the left of Table 2, the optimal optimization-matrix for calculating the quotation by job $a_{1,2,3,4}$ allocating resource r_1 is depicted.

Now assuming the time stamp T' and event 5 occurs. On the right of Table 2, the optimization matrix of job a_5 is depicted. Someone might mention that the micro-objectives 1 to 4 on the right of Table 2 are assigned to the idle resource

Table 1. Left: Optimization matrix in the second step. Right: Optimization matrix in the third step.

	micro-objectives n...N		micro-objectives m...M			
resources	1	2	1	2	3	4
r_1			X	X	X	X
r_2						
r_3						
$r_\varnothing$						
displaced						

	micro-objectives n...N		micro-objectives m...M			
resources	1	2	1	2	3	4
r_1			(X)	(X)	(X)	X
r_2			-	-	-	-
r_3	X	X	-	-	-	-
$r_\varnothing$					X	
displaced			X	X		

Table 2. Left: Optimization matrix to calculate the quotation by job $a_{1,2,3,4}$ on resource r_1. Right: Optimization matrix to illustrate how job a_5 takes over r_1.

	micro-objectives n...N				micro-objectives m...M	
resources	1	2	3	4	∅	∅
r_2			X	X		
r_3	X	X				
$r_\varnothing$						
displaced						

	...	micro-objectives m...M			
resources	5	1	2	3	4
r_1	X				
r_2		-	-	-	-
r_3		-	-	-	-
$r_\varnothing$		X	X	X	X
displaced					

$r_\varnothing$. It has to be considered that a_5 is not responsible for job $a_{1,2,3,4}$. To job a_5 it seems to be a virtual single-step optimization. $a_{1,2,3,4}$ takes care of the fallback resources by itself by calculating the quotation.

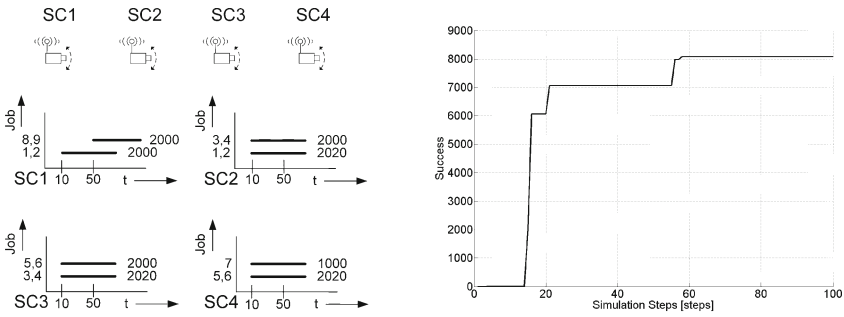
5 Related Work

In this section, the related work is discussed. First, the job-resource model is considered. The name of the model is influenced by scheduling theories [2]. The scheduling theory focuses on the amount of tasks and machines, their type and their arrival rate to classify the scheduling problem. The presented job-resource model is more general and focused on distributed smart systems. A job can be considered as a software agent, respectively intelligent agent, in multi-agent systems as described in [11]. The approach of using software agents in the execution of tasks is well established. In Monari et al. [5], an agent-based multi-sensor process for each object to be tracked in a smart camera network is created. An agent is focused on the execution of computer vision like data association and fusion. They used a well defined system architecture called NEST, which is still a research field [6]. Ukita et al. [9] also used agents. In their approach, the smart system components are the agents (so called active vision agents). Our focus is on the definition of a model that is as accurate as necessary and as less restrictive as possible. It is more general and defines interaction possibilities to achieve micro- and macro-objectives.

The second research objective of an optimization transparency and a system-wide optimization have not been considered in that way before. Most approaches used in the field of distributed smart camera networks make use of auction based algorithms like [7], which can be understood as a distributed greedy search. Also, some algorithms using negotiation chains were developed, for instance in [8]. The handling of these chains needs more maintaining than the proactive quotation-based approach we used. The heuristic expanding of this approach [3] enabled the parallel execution of tasks on smart components, which is a necessity to be able to track multiple persons by the same smart camera.

6 Evaluation

We presented an algorithm, which is capable to optimize the execution of multiple micro-objectives on resources. The optimization is transparent to a job and is especially a local single-step optimization from its point of view. Caused by the proactive quotation broadcasting, the optimization is done directly and it is avoiding negotiation chains. To show this, we evaluated the algorithm using the MASON[1] simulation toolbox. We set up a resource network as depicted in Fig. 8(a). Each resource is only capable to communicate with its direct neighbor resource. A message needs 3 simulation steps to be transferred from resource to resource. The graph in Fig. 8 shows the average system success at each simulation step, measured during the evaluation of 10 repeats. During the initialization phase step 10 to step ∼20, the optimization converges up to a success of 7000. At simulation step 50, the events 8 and 9 occur. The system needs only 10 steps to reach the maximum success and to reconfigure the whole network. Keeping in mind that the message exchange takes 3 steps.



(a) Evaluation setup

(b) System success at each simulation step.

Fig. 8. Simulation setup with resulting graph.

In the following, we want to demonstrate the capability to solve a complex problem, like the generalized assignment problem (GAP) [4] using only local algorithms. Therefore, we used a benchmark set[2]. The term $\max_{i',g_{sub},g'_{sub}}$ in Eq. 9 represents the computationally intensive part. This is handled by using stochastic tunneling (ST) in the third step of the heuristic PQB approach. We compare the results of stochastic tunneling implemented as central approach to the introduced local PQB heuristic. A single run of the central ST contains 500 explorations to find the best assignment. These runs were repeated 200 times within a single simulation. Each simulation (central ST and local PQB) was repeated 10 times. The results of the 200th step are depicted in Table 3. It is obvious that the ST heuristic is not the best approach to solve GAP. The evaluation also showed that the local

[1] http://cs.gmu.edu/~eclab/projects/mason/.
[2] http://people.brunel.ac.uk/~mastjjb/jeb/orlib/gapinfo.html.

PQB approach is able to reach similarly good results as a centralized approach with a percentual deviation of 5.37 % and 8.31 %.

Table 3. Comparison of central ST vs. local PQB

Benchmark set	Optimal solution	Ø central ST	Ø local PQB	Deviation
c515-1	336	285.0 (stdev:5.88)	269.7 (stdev:11.55)	5.37 %
c515-2	327	283.9 (stdev:5.37)	260.3 (stdev:17.69)	8.31 %

7 Conclusion and Future Work

In this paper, we described a holistic approach for task execution in distributed systems. Therefore, a job-resource model was introduced. This model pursues individual objectives (micro-objectives), which might be created by a user, as well as system-wide objectives (macro-objectives) considering e.g. load-balancing issues. The optimization of macro-objectives is transparent to a job. More precisely, to a job it appears as a so-called virtual local single-step optimization (VLSO). This is achieved by means of the presented technique called proactive quotation broadcasting (PQB). In the presented algorithm, micro-objectives are able to share resources. The challenge of grouping objectives on resources, and an approach to overcome it, was explained. Experimental results revealed the capabilities of the PQB algorithm, i.e. its fast reaction to disturbances and its ability for an optimal objective-to-resource assignment.

Future research activities will focus on using alternative heuristics for the third step of the algorithm, e.g. evolutionary strategies, instead of stochastic tunneling.

References

1. Bramberger, M., Doblander, A., Maier, A., Rinner, B., Schwabach, H.: Distributed embedded smart cameras for surveillance applications. Computer **39**(2), 68–75 (2006)
2. Conway, R.W., Maxwell, W.L., Miller, L.W.: Theory of scheduling (1967)
3. Jaenen, U., Spiegelberg, H., Sommer, L., von Mammen, S., Brehm, J., Haehner, J.: Object tracking as job-scheduling problem. In: 2013 Seventh International Conference on Distributed Smart Cameras (ICDSC), pp. 1–7, October 2013
4. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. Wiley, New York (1990)
5. Monari, E., Maerker, J., Kroschel, K.: A robust and efficient approach for human tracking in multi-camera systems. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 134–139, September 2009
6. Mossgraber, J., Reinert, F., Vagts, H.: An architecture for a task-oriented surveillance system: A service- and event-based approach. In: 2010 Fifth International Conference on Systems (ICONS), pp. 146–151, April 2010

7. Rinner, B., Dieber, B., Esterle, L., Lewis, P., Yao, X.: Resource-aware configuration in smart camera networks. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 58–65, June 2012
8. Starzyk, W., Qureshi, F.Z.: A negotiation protocol with conditional offers for camera handoffs. In: Proceedings of the International Conference on Distributed Smart Cameras, ICDSC 2014, pp. 17:1–17:7. ACM, New York (2014)
9. Ukita, N.: Real-time cooperative multi-target tracking by dense communication among active vision agents. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 664–671, September 2005
10. Wenzel, W., Hamacher, K.: A stochastic tunneling approach for global minimization. Phys. Rev. Lett. **82**(15), 3003–3007 (1999)
11. Wooldridge, M.: An Introduction to Multiagent Systems, 2nd edn. Wiley, Chichester (2009)