Workflow Management Systems + Swarm Intelligence = Dynamic Task Assignment for Emergency Management Applications

Hajo A. Reijers[1], Monique H. Jansen-Vullers[1], Michael zur Muehlen[2],
and Winfried Appl[2]

[1] Eindhoven University of Technology, Department of Management Technology, Den Dolech 2, 5600 MB, Eindhoven, The Netherlands
{h.a.reijers,m.h.jansen-vullers}@tue.nl
[2] Stevens Institute of Technology, Howe School of Technology Management, Castle Point on Hudson, Hoboken, NJ 07030, USA
{michael.zurmuehlen,winfried.appl}@stevens.edu

Abstract. The assignment of tasks to human performers is a critical component in people-centric business process management systems. Workflow management systems typically assign work items using strategies that only consider qualified resources. There are, however, situations, where this approach falls short. For instance, in emergency response situations, tasks need to be carried out by resources that are available immediately, even if they do not match all skill requirements. This paper compares the performance of a set of six task assignment mechanisms for workflow applications using a scenario from the emergency management domain. In particular, we develop and simulate assignment strategies inspired by stimulus/response models derived from swarm intelligence, and benchmark these strategies against conventional task assignment strategies. Our findings show that swarm intelligence-based approaches outperform the traditional assignment of tasks in ad-hoc organizations, and that workflow-based emergency management systems could benefit significantly from these novel task assignment strategies.

Keywords: Business Process Management, Workflow, Task Assignment, Swarm Intelligence.

1   Introduction

Workflow Management Systems (WfMS) coordinate tasks, resources and data according to the formal representation of the process logic, the workflow model [1]. The assignment of work items to human performers is a critical component in people-centric business process management scenarios. Excessive task automation and poor design of work assignment strategies are critical issues that can jeopardize the success of workflow projects [2].

During the *build time* of a workflow application, the workflow application designer has to describe both the structure of the business process to be automated, and the

resources that carry out the process. At *run time*, work items are assigned to resources based on assignment policies that determine the strategy for how process work should be allocated. Upon the instantiation of a workflow task, the workflow enactment service places work items on the work lists of qualified performers who are determined using a process of *role resolution.* For the assignment of pending work items different strategies can be implemented, such as first-come-first-served, market-based allocation mechanisms or hierarchical distribution algorithms. These strategies have an impact on how the workflow enactment service prioritizes activities and notifies candidate performers.

Many commercial WfMSs focus on process routing, while (1) oversimplifying resource and task attributes [3] and (2) providing limited facilities to represent dynamic changes in the actual organizational structure of their environment [4]. The factors used to determine the optimal set of resources to be charged with a pending activity is currently workflow-driven: qualifications of resources are treated as *static* values. The amount of dynamics employed in current WfMSs goes no further than linking the allocation mechanism to certain properties of the process instance at hand, e.g. its priority.

In this paper we focus on the domain of Emergency Management Services, where the shortcomings of traditional workflow-based task assignment mechanisms become very apparent. Emergency Management Services are concerned with improving public safety, and share the common objective of responding to citizen calls for assistance as quickly as possible to reduce loss of life and injury [5]. Examples of these services are those delivered by police and fire departments and emergency medical services of hospitals. Characteristic for emergency settings is that after a work item has been available for some time, it should rather be performed by a less qualified resource, than not performed at all (e.g., first response in case of injuries).

As a source of inspiration to extend workflow assignment policies, we turn to Swarm Intelligence [6]. This is a term dubbed for the collective behavior that emerges from groups of social insects. Social insects, such as ants or wasps, divide labor amongst the resources in such a way that the ratios of workers performing different tasks can vary (i.e., workers switch tasks) in response to internal perturbations or external challenges. Algorithms that mimic this behavior have been successfully applied to reduce set up times and throughput times for production scheduling in industrial settings (e.g. [7] and [8]).

The paper is organized as follows. In the next section, we introduce the sub-area of Swarm Intelligence relevant to our study, the stimulus/response model by Bonabeau et al. [9], and discuss the specific requirements of the emergency management domain. Section 3 outlines our research design: We use discrete event simulation to evaluate the effectiveness of various task assignment strategies in a realistic emergency management scenario. The results are presented in Section 4, which is followed by an overview of related work (Section 5). We conclude the paper with a discussion of our findings, limitations, and an outlook on future work (Section 6).

2 Background

2.1 Swarm Intelligence

For years, scientists have been studying ants, bees and wasps because of the amazing efficiency of social insects in finding the shortest path to a food source, spreading alarm in a colony, or dividing labor [6]. Building on many empirical experiments and observations, various models for labor division of social insects were developed (e.g. [10]). These so-called *threshold models* consist of two components. First, a *threshold* exists for each resource towards each task type, which indicates how responsive a resource is towards a certain task type. The lower this threshold is, the more responsive a resource becomes to perform a task of this type. The second component involves the *stimulus*, which is available for each pending task. The more important a task becomes, the higher its stimulus will be. Eventually, even resources with a high threshold towards a certain task type will respond to a work item of this type, given that it has a high stimulus.

In this paper, threshold models are modified to show behavior similar to bidding mechanisms for task assignment. The setting is a workflow environment where pending work items are approached similar to a job market with job seekers of different activity levels. Job seekers with a low task threshold will make a relatively high bid and the highest bidder will be given the pending task, while job seekers with high threshold levels will not become active until the stimulus reaches the threshold (compare [9]). The threshold for each job seeker to perform a certain task at a particular time scales with the contribution that a job seeker adds to the global performance. For example, the threshold may relate to the shortest distance for a wasp to pick up food [10,11], so that the global optimum is a minimum function, i.e. the overall time that is required to pick up all food should be minimal.

Threshold models can be extended with a learning mechanism. This mechanism ensures that the threshold towards a certain task type decreases when a resource is working on that task type (i.e., the resource learns to perform it well) and increases for all other resources that are not performing that particular task type (i.e., they forget how to perform the task). The threshold models that include the learning mechanism are referred to as *learning threshold models*, in contrast to the *fixed threshold models*. One advantage of a learning threshold model over a fixed threshold model is in the area of robustness, which various biological studies point out as an essential element of colonies [12,13].

In this study, we consider three different threshold models, which we introduce more formally now. We denote a threshold with $\pi_{r,i}$, which represents the threshold for resource r towards task i at a certain time. The stimulus S_i describes the demand to perform task i at a particular time. The stimulus that is used for our task assignment studies is updated after each discrete time step with a constant δ. This way, the stimulus S_i is used to improve the probability of completion for task i, which becomes more important over time [14]. The fixed threshold model for ants (F-ANT), as proposed by [15], lets resource r bid for task i with the following bid:

$$\frac{S_i^2}{S_i^2 + (\pi_{r,i}^2)^\chi},$$

where $\chi \in [1,\infty)$ is a moderating coefficient that determines the impact of the threshold on the level of the bid. If two resources both place the highest bid, the task is allocated randomly between these two resources.

A specific case of the F-ANT model is the fixed threshold model for wasps, as described in [10]. Here, $\chi=1$. We refer to this model as F-WASP.

We also consider a learning threshold model, which can be seen as a refinement of the F-WASP model. An additional threshold $\theta_{r,i}$ is introduced, which exists for resource r towards task i and which evolves over time. Using ξ and φ as respectively the learning and forgetting coefficients, $\theta_{r,i}$ is lowered at each discrete time step with ξ when a resource works on a task of type i and increased with φ if not. In the learning threshold model for wasps (L-WASP), resource r bids for task i with the bid:

$$\frac{S_i^2}{S_i^2 + \alpha\theta_{r,i}^2 + \beta\pi_{r,i}^2},$$

where α and β are positive coefficients that determine the relative importance of the thresholds and θ is usually restricted to a certain positive domain.

The F-ANT, F-WASP, and L-WASP models can be considered as the basic threshold models in the Swarm Intelligence domain and they were chosen based on their known value in industrial optimization problems (e.g. [7] and [8]).

2.2 Emergency Management Domain

The performance of typical business processes, such as those found in governmental agencies, banks, insurance companies, etc., is measured in a variety of ways. This variety stems from the different stakeholders involved that may pursue different interests. However, in the Emergency Management Domain the primary concern is to reduce loss of life, injury and damage to property. Therefore, *timeliness of execution* is the most dominant performance evaluation criteria in this domain [5]. Other considerations such as efficiency and costs are often irrelevant, neglected, or can be seen as variations of the time criteria. In this study we focus on two different ways of making this criteria operational:

1. *Throughput Time* (TPT), which measures the time between an incoming Emergency Call and the moment that the incident is resolved, and
2. *Response Time* (RT), which is the time between the incoming Emergency Call and the moment that the emergency response begins at the location of the incident.

While an incident's TPT includes its RT, favoring one criteria over the other may lead to different decisions. Consider, e.g., the dilemma to send out an available unit to assist at a large incident X (to which a single unit has already responded) or to respond to a small incident Y that has just occurred. The first option will lower the TPT of X, the second will lower the RT to Y. While minimizing TPT reduces the negative consequences of incidents, such as loss of life, injury and damage to property, a large RT has its own set of negative consequences [16,17]. Just as described in [16], we consider TPT and RT as equally important.

Furthermore, even though reducing the average TPT and RT is of the utmost importance, emergency responders must take into account the principle of *equity* [5],

which states that similar incoming emergency calls must be treated equally. In particular, incidents that occur further away from an emergency center must not be structurally neglected in favor of nearby incidents.

Typical business processes differ from emergency services in that resources in the latter environment are generally trained and equipped to perform all kinds of tasks besides their specialization, while this is not necessarily so in non-emergency settings. The advantage of this generalist approach is that non-specialists can perform tasks when specialists are unavailable. The disadvantage of this approach is that the processing times for tasks performed by non-specialists are typically greater. For example, a fire-fighting unit with a small aerial ladder is capable to perform high angle rescues using ropes and manual ladders. Characteristically, this takes longer than a rescue attempt by a team using a vehicle with a longer and flexible aerial ladder [16]. As we will demonstrate, a supply of heterogeneous resources is an important ingredient to the emergency response scenario that we use to benchmark different task assignment strategies.

3   Methodology

In this section we benchmark the task assignment mechanisms that originate from the swarm intelligence concepts with three conventional task assignment mechanisms, using an example scenario from the emergency management domain. First we describe the three conventional assignment mechanisms used for benchmarking. We then introduce the simulation scenario, followed by the simulation model, the simulation approach and the design of the experiments.

3.1   Conventional Task Assignment Mechanisms

To better understand the performance and the behavior of the threshold models introduced above, we evaluate them against three conventional task assignment mechanisms: First-in, First-out (FiFo), a Greedy dispatch rule and the Dynamic Model.

The FiFo mechanism assigns tasks in the sequence of arrival of new cases. FiFo queuing is a simple and robust allocation rule [18] and widely used in commercial WfMSs [4]. Tasks are dispatched based on a best-available basis. If no qualified performer can be found in the system, the assignment of a task will be deferred until a qualified resource becomes available. Most WfMSs buffer this gap by using work lists as local queues for individual resources. The FiFo mechanism will then place arriving tasks on the work list of (one or more) suitably qualified resources. The actual allocation of work (i.e., the decision, which of multiple resources performs the task) can be implemented using a similar First-Come-First-Served mechanism, or through auction or other bidding protocols.

The Greedy mechanism assigns tasks to resources that can complete the task in the shortest time possible. This heuristic has been applied to several task assignment problems [8,19,20]. In a WfMS the Greedy mechanism resembles the Shortest Processing Time (SPT) rule, which can be used as a dispatch rule in WfMSs [18]. The SPT rule optimizes the assignment of pending tasks based on the assumption of

resource-independent processing times of the task. The Greedy rule ensures that a pending task is assigned to the resource that guarantees the shortest processing time. Since in practice task-processing times depend on the capabilities of individual resources, a Greedy mechanism needs to compute the expected completion time for all task-resource combinations that qualify for the assignment.

The third model used as a benchmark for the threshold approach is a task assignment model proposed by Kumar et al. [3]. This model includes the parameters *suitability* and *urgency*. Suitability is the inherent qualification of a resource to perform a specific task. This may include qualifications, authorizations, and permissions. In addition, each work item is assigned a time-dependent urgency value. Each of the resources (r) bids for a work item (i) using an assignment function taking into account *suitability (r,i)* and *urgency(i)*. The work item is then assigned to the resource with the highest computed bid. Note that the other two parameters in Kumar's model are not considered in this study, as they add little value in the scenario under consideration (see Section 3.2). In particular, there are no constraints, which makes the *conformance* parameter obsolete, and resources work in shifts, so that the *availability* parameter is not adding much value. While each of the three benchmark mechanisms represents a dynamic assignment mechanism, we denote Kumar's model as the Dynamic Model, just as it is referenced in the original work.

3.2   Emergency Response Scenario

To benchmark all proposed task assignment mechanisms we performed a simulation study using a fictional emergency response scenario. This scenario contains a Local Fire Station (LFS), which responds to relatively small incidents in its district, i.e. Emergency Calls (ECs). Such emergencies never require more than three fire fighting units. The process descriptions, resources and processing times for this scenario are based on data from the Austin Fire Department [16].

In this scenario, we defined a limited set of twelve EC types that have equal priority and a set of heterogeneous resources with different specializations (i.e., varying levels of task suitability). An EC of type 'water rescue' requires a specialized rescue team with diving skills. Should this team be unavailable, a less suitable resource can (and should) respond to the incident, e.g., one fire fighter with a boat may respond. If a non-perfect resource responds, the processing times of incident-related tasks will increase. The resources work in shifts of 24 hours. Within these 24 hours all personnel is available for work. After 24 hours the shift personnel is replaced by a new set of resources. There are always enough resources available to fill a shift, i.e. we do not account for absences or vacation times [16]. When maintenance is performed, a spare vehicle with exact the same specification is available.

Processing times are based on a defined minimal processing time, which may be different for each type of tasks. If a task is assigned to a less suitable resource, this minimal processing time is multiplied by a penalty factor that accounts for the degree of unsuitability.

A Location Model defines the operating area of the LFS. This Model is used to compute the travel distance of units to incidents within the LFS district. The distance between the responders and the incident is an input parameter for the task assignment mechanisms. The calculation of distances is done using Euclidean or Manhattan

distance grids [21]. The Location Model is a square shaped territory (grid) that indicates a part of a city. At the north side of the territory a river runs from east to west. The LFS is located at the center of the territory.

3.3   Model Building and Validation

All models used for the experiments share the same structure and were built using CPN tools. This simulation language is based on the logic of Colored Petri Nets and is suitable for discrete event simulation [22].

All models share a six module structure as shown in Fig. 1. The *environment module* represents the environment of the LFS. This module creates the ECs and defines their associated characteristics. Newly arriving ECs are routed through the model and are handled in the *process module*. When resources have finished working on the tasks for a particular EC, they will still be located at the incident site. The *drive back module* manages their return to the LFS. Each time a work item is made available by the process module it is managed by the *task module*. This module queues the task until the task is assigned and completed. After the task has been completed, the task module allows the process module to access the task again. The task module consists of two other modules, i.e. the *failure module* and *allocation module*. The allocation module is unique for each proposed mechanism.
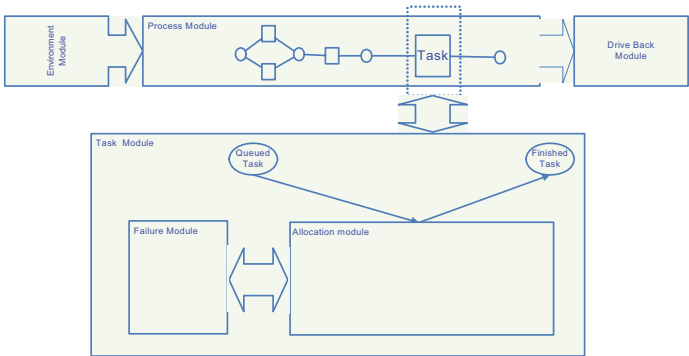

Fig. 1. General model structure

The CPN models are used to collect data, and to analyze the performance of the different assignment mechanisms in the sample scenario with regard to TPT and RT. Data is collected for each EC type individually. We are interested to learn whether mechanisms that seem to perform well on overall TPT and RT treat individual EC types different from other mechanisms. The average utilization of each resource is measured for all experiments performed in this study.

Verification of the model consists of checking the code, inspecting output reports and verifying that the modeled elements correctly represent the real world equivalents [23]. Making use of the state space tool, it is possible to check the model on home, liveness and fairness properties [22]. In addition, we verified the model by simulating EC distributions and routing. The expected counts were within a 99% confidence interval of the observed CPN model counts, hence, we considered the CPN model to

be qualified. Since the model is based on a fictional case it is not possible to validate the model against real life data or historical data. The use of mathematical queuing models is not feasible in this situation, because processing times heavily depend on the resource that performs a task. For this reason a simulation study is essential.

3.4   Simulation Approach

Prior to starting the actual experimentation stage, a number of issues need to be addressed, i.e., the warm-up period, run length and the number of replications. Also, a number of parameters must be set in order to optimally use the mechanisms.

The length of the warm-up period needs to be evaluated, if the state of the model at starting time does not represent the steady state of the actual system. The warm-up period is the amount of (simulated) time that a model needs to run before the statistical data collection begins [23]. In this research, the warm-up period is evaluated based on the moving average of the TPT. It appeared that all experiments that are concerned with non-learning assignment mechanisms, i.e., FiFo, Greedy, the Dynamic Model and F-WASP, evolve in the same way, and reach a steady state at t =25000 minutes. For the L-WASP mechanism the steady state is observed from t =36000.

Once the model has warmed up, the run length of the model has to be decided. One method for deciding this is inspecting the random numbers sampled. As a rule of thumb, a minimum of 15 to 20 random numbers for each type of random number stream should be used in this model. To ensure that this takes place for all of the random number streams the least frequent event in the simulation model should be selected and the model should be run until this event occurs 15 to 20 times [23]. This event is the occasion of a chemical structural fire with a probability of 0.011. Therefore, at least 1818 ECs need to be observed to detect 20 chemical structure fires of this category. Based on the lowest arrival rate under consideration, i.e., λ=11000, a period of t = 86867 minutes has to be considered. In this study a run length of t = 125000 was applied, which is approximately a three month run length.

Due to the very nature of random numbers, it is imprudent to draw conclusions from a model based on the results generated by a single model run. Replication is defined as executing the same model a number of times n, but with different random numbers in each run [23]. A statistical method for determining the number of replications is described by [24]. Applying this procedure led to a satisfying accuracy level at 20 replications for all mechanisms at low and high arrival rate, i.e., the deviation of a replication never exceeded 5% of the average.

In addition to the general simulation settings, some parameters need to be tuned for the Dynamic Model and the threshold models. For the Dynamic Model two parameters have to be tuned: urgency (u) and suitability (S), which are both in the interval [0.0-1.0]. The Dynamic Model originally uses an urgency level of interval [0.9-1.0] with parameter u =0.1 when a work item is queued for a period t. This rigid approach is refined in this study. A larger interval is chosen, i.e. [0.5-1.0] and u =0.05: The urgency level starts at 0.5 and increases each discrete time step with a constant 0.05. The suitability parameter S is also in the interval [0.5-1.0]: The least capable resources to perform a task type have a suitability rating of 0.5, the best one for that same task type has a suitability rating of 1.0. The rest of the capable resources scale in between this interval.

The settings of the threshold models are based on the literature and a sensitivity analysis. The stimulus (S) represents the number of time steps an EC is in the WfMS, in this study we use S=1 [14]. The parameter tuning for the learning mechanism can be based on a genetic algorithm [25], a simple hand tuning technique [26], or a sensitivity analysis that first sets the most important parameter followed by the parameter with the second highest impact, and so on [27]. Based on the latter approach, we derived the threshold values as well as the learning importance α and the task duration component β. The exact values have been tuned based on a sensitivity analysis. For L-WASP this resulted in α=0.02 and β=1; for F-ANT in χ=1.1. The learning coefficient ξ and forgetting coefficient φ have an insignificant or moderate influence and tuning is not necessary. To set these parameters in a sensible manner in this study, the values from the social insect behavior are taken: ξ=10 and φ=1 [10].

3.5   Design of the Experiments

In this study, we compare six different mechanisms for task assignment. We want to answer three distinct questions. How do the mechanisms perform regarding the TPT and RT:

1.   At increasing arrival rates?
2.   When the fire station is located further away from the river (both at low and high resource utilization levels)?
3.   At an increasing failure rate (both at low and high resource utilization levels)?

To answer the first question we used six different scenarios. The arrival rate is denoted by λ and represents the average arrival of ECs each year. We chose arrival rates that correspond to an average resource utilization of the mobile resources of respectively 0.34, 0.43, 0.53, 0.62, 0.71, and 0.80. The six alternative mechanisms are tested for these six different arrival rates. This results in 36 experiments.

To answer the second question we set two parameters to derive the different scenarios. For each of the two settings two different *river locations* are chosen, thus influencing the traveling time. In this alternative, also two different *arrival rates* are considered, i.e. for a low and high resource utilization. The two settings of the two parameters result in four scenarios.

To answer the third question, we again set two parameters to derive the different scenarios. For each of the two settings two *failure probabilities* are proposed: 0.02 and 0.05. Again, two different *arrival rates* are considered. The two key parameters with two settings results in four scenarios.

In total over a hundred experiments were performed to address the objectives of this simulation study. The complex CPN models require a lot of computing power. Experiments that require models with high arrival rate settings (23000 arrivals a year), take approximately 15-30 hours to finish all 20 replications on a Pentium 4 with 5GB RAM. Five such systems were required for a period of three weeks to perform all experiments. The output of all these experiments were collected, analyzed and documented according to the structured procedures as described in the next section.

3.6 Procedure for Mechanism Comparison

The performance of each mechanism during a particular experiment depends on RT and TPT that are measured for all ECs and for each EC type individually. To compare different alternatives the procedure mentioned below was followed.

For each experiment, a summary of all 20 runs for each performance was provided in one overall CPN report, including the average and the standard deviation of all 20 runs. To assess the performance for all mechanisms, the overall RT and TPT were plotted in a graph for the increasing arrival rates. To test whether the differences between mechanisms are significant a pair wise comparison was made. It is not safe to assume equal variances, therefore we applied the Welch test and not a pooled variance test 24]. When comparing more than two alternatives and making several confidence interval statements simultaneously, the individual confidence levels of the separate comparisons have to be adjusted upwards to reduce the number of type 1 errors (rejecting the null hypothesis when it is true). Therefore, we applied the Bonferroni correction to all measurements [24,28]. To test whether a specific mechanism respects the equity property (similar incoming emergency calls are treated equally), a Kolmogorov-Smirnov test was applied to determine whether the emergency incidents that exceeded the response time limit were uniformly distributed over the Location Model.

4 Results

4.1 Ranking

On the basis of the simulation study, the various allocation mechanisms can be ranked with respect to their performance in minimizing TPT and RT. For each of these criteria, a mechanism is ranked higher if its average value as aggregated over all ECs is significantly lower. If two mechanisms do not differ in this respect, the mechanism that significantly outperforms the other for *most* of the 12 EC types is considered to be better. The latter procedure was explicitly necessary to distinguish between the performance of the F-ANT and F-WASP mechanisms. For all comparisons, a confidence level of 95% is applied.

In Fig. 2, mechanism rankings are shown for low and high levels of utilization, i.e., aggregated over the lowest and highest three levels of resource utilization.

Our analysis provides various insights, of which the most important ones are as follows:

- At both utilization levels, the Greedy mechanism delivers the best performance with respect to RT. However, it is the only mechanism that violates the equity property (this is indicated by the shading in Fig. 2). The Greedy mechanism leads to a favorable handling of incidents that are closest to the LFS, since it prefers the use of idle resources for performing tasks with a short task duration which *includes* the travel time. Clearly, this is unacceptable for society (e.g. cats are saved from nearby trees while the chemical factory further away is burning).
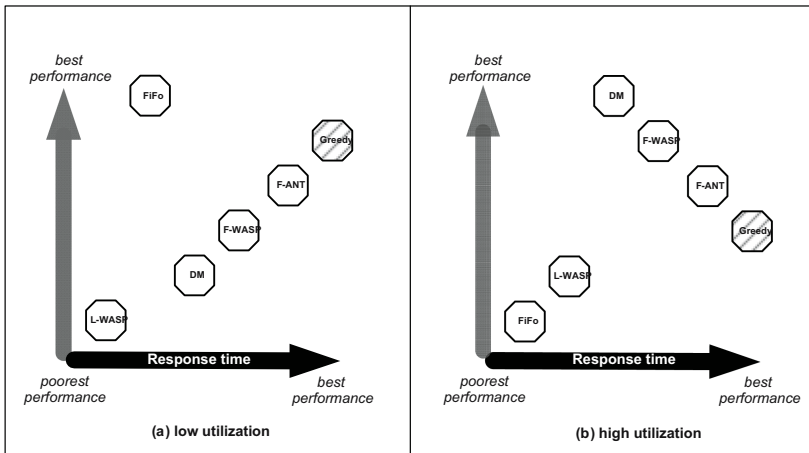
Fig. 2. Ranking of allocation mechanisms

- In contrary to the good performance of the L-WASP mechanism in studies like [8], [9] and [29] it performs markedly poor in this study. The reason for this is that its learning property has a negative impact in the context of the heterogeneous fire-fighting units. The incidental assignment of EC types to less suitable resources (e.g. when more suitable resources are occupied) leads to a structural preference for using such resources over time.

- The FiFo mechanism performs well with respect to TPT when resource utilization is low, but it is the worst performing mechanism overall at high utilization levels. Because it strictly focuses on the arrival pattern, it completely fails to exploit the different resource capabilities.

- The Dynamic Model is almost the mirror image of the FiFo mechanism. Its performance is poor at low levels of utilization, but it is the best performer on TPT at high utilization levels. The reason for the latter is its subtle balancing of the incident urgency and a resource's suitability, while ignoring the task duration. We found that the Dynamic Model commits particularly fast to EC types that multiple resources can work on, while other mechanisms (e.g. Greedy, F-WASP and F-ANT) prefer work items that they can finish fast (at least initially). Because the Dynamic Model does not consider the task durations, it is outperformed with respect to response time at high utilizations by the F-ANT and F-WASP mechanisms.

- The F-ANT and F-WASP mechanisms provide the best trade-offs in minimizing both TPT and RT. When not considering the Greedy algorithm (violating the equity property) they are only outperformed at TPT. At a high utilization level it depends on the relative value of minimizing either TPT or RT which of the two mechanisms is preferable. Note that the absolute differences in TPT and RT between the two are generally smaller than for other mechanisms.

- The *absolute* differences in TPT and RPT between the various mechanisms tend to increase when the utilization increases. Clearly, this is not captured by our ranking of the models in Figure 2, as the axes reflect ordinal scales. There is no

satisfactory, way to summarize the absolute differences within the limitations of this paper because of the great variations of these differences over EC types and the statistical subtleties that result from aggregating these differences. The most important insight here is that at higher utilization levels the choice of assignment mechanism is all the more important, as differences become more apparent. In other words, with an excess of resources everything will turn out well anyway.

4.2 Robustness

To test the robustness of the rankings as presented in the previous section, we examined two additional scenarios (see Section 3.5). In the first scenario, the river is relocated from the northern side of the grid to the center, close to the LFS. We expected this to generate a general advantage with regard to RT and TPT for this EC type (e.g. river rescues) of 3.068 minutes. Oddly, the gap between the L-WASP mechanism and the other mechanisms turned out to be larger after the relocation, perhaps because of the poor allocation decisions it makes anyway. In addition, FiFo and the Dynamic Model improve their performance after relocation, which conforms precisely with the expected gain. This makes sense: Both mechanisms do no consider travel time in their decision-making. For Greedy, F-ANT and F-WASP mechanisms, their favorable position increases towards the other mechanism for the river rescues. But only for the Greedy mechanism this increase is significantly larger than expected. This once more illustrates that this mechanism structurally favors incidents that are close to the LFS.

In the second scenario, the impact of an increasing (mechanical) failure rate of firefighting units was examined. From the evaluation of this scenario we conclude that the difference between the Dynamic Model on the one hand and F-WASP and F-ANT on the other decreases as the failure rate increases. Also, the L-WASP mechanism performs even poorer when the failure rate increases. The overall ranking of the mechanisms, however, is not affected. In summary, the evaluation of both scenarios suggests that the ranking incorporates a certain level of robustness.

5 Related Work

Decentralized resource allocation is of particular interest to various scientific domains. An impressive amount of studies has exposed different aspects of the problem and respective algorithms for solving it. We will subsequently point out only a small excerpt of these approaches which are most relevant to our research.

In the workflow management domain, numerous authors tried to tackle the problem from an implementation perspective [30,31]. They predominantly focus either on modeling organizational structures with process elements linking to them, e.g. [4,31,32] or on the definition of criteria for assignment mechanisms, e.g. [33,34]. Research in the area of resource management in workflow applications is centered around access control mechanisms and policies that permit or restrict the ability of individual resources to perform tasks [35,36]. Dynamic resource allocation is of considerable importance to the fields of distributed (grid) computing [37], robotics [38] and multi-agent systems [39]. These domains draw heavily on market-based

algorithms, as well as on reinforcement learning techniques. Shen proposed to extend current task assignment mechanisms beyond the static role-concept by including criteria such as the social proximity of workflow participants, or the compatibility of tasks with the existing content of work lists [34]. Allocation mechanisms inspired by economic principles, such as auctions and games, have been studied by Tan and Harker [40], as well as Alt et al. [41]. Auction protocols for the scheduling of decentralized resources have been discussed in other domains as well [42].

Despite the considerable amount of related work, we are unaware of research that specifically deals with dynamic algorithms for distributed task assignment in the business process management domain. There are two notable exceptions, however: the model by Kumar et al [3] takes into account the tradeoff between flexibility and efficiency (see Section 3.1). For the example case described therein, we were able to produce similar results with our stimulus/response model (these results are not included in this paper because of page restrictions). This indicates a high similarity in the effectiveness of both approaches. However, in this paper we show that Kumar's model is less robust in dealing with different time criteria.

The other exception is the work in [43]. This approach is based on the estimation of execution times and possible routes that cases will follow. When a new case arrives, a snapshot of the system is taken and a static scheduling problem based on this snapshot and the estimations is being solved. The resulting preliminary schedule is implemented and the whole procedure is repeated as soon as the next job arrives. The capability of this algorithm to minimize late jobs depends on the accuracy of the estimations and the solution quality of the scheduling instance. An important insight from this paper coinciding with ours is that when utilization rates are greater than or equal to 65%, almost every other technique than the FiFo rule is advantageous.

6 Discussion and Conclusion

Our studies show a favorable performance of fixed stimulus/response-models as a basis for workflow task assignment in emergency response situations. In particular, these models provide a balanced trade-off between the performance criteria that are important in this domain.

Among the most important limitations of our study, it must be noted that in our simulation model the execution of tasks cannot be interrupted: ACID properties are strictly enforced [18]. In real world fire fighting situations, tasks can be interrupted (e.g., a fire fighter will stop saving a cat from a tree if called to extinguish a chemical fire elsewhere). Also, we do not consider false alarms, which sometimes amount to 50% of all fire incidents [16]. Finally, the higher utilization levels we studied are beyond what is normal for emergency response situations (there, utilization levels are typically between 35% and 45% [44]). These limitations restrict general statements on the effectiveness of workflow technology in the emergency management domain. At the same time, various developments point at the increasing importance and use of workflow technology in this domain, as illustrated by the RESCUE [45], CITI [46], and AMIRA projects [47]. In addition, the emergency management domain itself may undergo changes, e.g., in the form of increasing resource utilization at local fire stations and by assigning larger incidents to regional fire stations. The implementation

of swarm intelligence-based algorithms could contribute to robust process performance both at low and high resource utilization levels.

The main insight from our study is that stimulus/response-based task assignment mechanisms are appropriate in environments where timeliness of call resolution is critical. In particular, as most WfMSs dispatch work items to their performers on a FiFo basis, the latter strategy should be reconsidered in situations where timeliness is critical and resources are scarce. Models inspired by swarm intelligence could serve as a template for mechanisms that are more sensitive to (a) the impact of the elapsed time on the urgency of cases and (b) variations in the suitability of cross-trained resources. Since the environment in which businesses operate is increasingly complex, it can be expected that Business Process Management will have to provide capabilities similar to those of emergency management systems in order to continually provide valuable competitive advantage.

References

1. WfMC Terminology and Glossary, 3rd edn. Workflow Management Coalition, Winchester (ID) (1999)
2. Moore, C.: Common Mistakes in Workflow Implementations. Giga Information Group, Cambridge, MA (2002)
3. Kumar, A., van der Aalst, W.M.P., Verbeek, H.M.W.: Dynamic Work Distributio. In: Workflow Management Systems: How to Balance Quality and Performance, Journal of Management Information Systems 18(3) (2002)
4. Zur Mühlen, M.: Organizational Management in Workflow Applications – Issues and Perspectives. Information Technology and Management 5, 271–291 (2004)
5. Swersey, A.J.: The Deployment of Police, Fire and Emergency Medical Units. In: Pollock, S.M., et al. (eds.) Handbooks in OR&MS, vol. 6, Elsevier Science, Amsterdam (1994)
6. Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems. In: Proc. NATO Advanced Workshop on Robotics and Biological Systems, Il Ciocco, Tuscany, Italy (1989)
7. Meyer, C., Bonabeau, E.: Swarm Intelligence: A Whole New Way to Think About Business, Harvard Business Review, May 2001, pp. 106–117 (2001)
8. Ghizzioli, R., Nouyan, S., Birattari, M., Dorigo, M.: An Ant-Based Algorithm for the Dynamic Task assignment Problem, TR/IRIDIA (2004)
9. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems, Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, Oxford (1999)

10. Bonabeau, E., Sobkowski, A., Theraulaz, G., Deneubourg, J.L.: Adaptive task allocation inspired by a model of division of labor in social insects; Biocomputation and Emergent Computing. World Scientific, Singapore (1997)
11. Price, R., Tino, P.: Evaluation Of Adaptive Nature Inspired Task Allocation Against Alternate Decentralized Multi-agent Strategies. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN VIII. LNCS, vol. 3242, pp. 982–990. Springer, Heidelberg (2004)
12. Robinson, G.: Regulation of the division of labor in insect societies. Annual review, Entomol 37, 637–665 (1992)
13. Wilson, E.: The sociogenisis of insect colonies. Science 228, 1489–1495 (1985)
14. Theraulaz, G., Bonabeau, E., Deneubourg, J.: Response threshold reinforcement division of labour in insects societies. In: Proceedings Royal Societies, London (1998)
15. Campos, M., Bonabeau, E., Theraulaz, G., Deneubourg, J.: Dynamic Scheduling and Division of Labor in Social Insects. Adaptive behaviour 8, 83–92 (2001)
16. Austin Fire Department: Austin Fire Department, Austin City Connections (2006), http://www.ci.austin.tx.us/fire/default.htm
17. Centraal Bureau voor de Statistiek, Brandweerstatistiek 2004; Voorburg/Heerlen (2004)
18. van der Aalst, W.M.P., van Hee, K.: Workflow Management: Models, Methods and Systems. The MIT Press, Cambridge, Massachusetts (2002)
19. Gerkey, B., Mataric, M.: Multi-Robot Task assignment: Analyzing the Complexity and Optimality of Key Architectures. In: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan (2003)
20. Gottlieb, J., Puchta, A., Solnon, C.: A Study of Greedy, Local Search, and Ant Colony Optimization Approaches for Car Sequencing Problems. In: Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.-A., Middendorf, M. (eds.) EvoIASP 2003, EvoWorkshops 2003, EvoSTIM 2003, EvoROB/EvoRobot 2003, EvoCOP 2003, EvoBIO 2003, and EvoMUSART 2003. LNCS, vol. 2611, Springer, Heidelberg (2003)
21. Rushton, G.: Applications of Location Models, Annual of Operations Research 18 (1989)
22. Jensen, K.: Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use, 2nd edn., vol. 1. Springer, Heidelberg (1997)
23. Mehta, A.: Smart Modeling – Basic Methodology and Advanced Tools. In: Proceedings of the 2000 Winter Simulation Conference (2000)
24. Law, A.M., Kelton, W.D.: Simulation Modeling and Analysis, 3rd edn. MCgraw Hill International Series (2000)
25. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)
26. Cicirello, V., Smith, S.: Wasp-like Agents for Distributed Factory Coordination. Autonomous Agents and Multi-Agents Systems 8, 237–266 (2004)
27. Kittithreerapronchai, O., Anderson, C.: Do ants paint trucks better than chickens? Markets versus response thresholds for distributed dynamic scheduling. In: Proceedings of the 2003 IEEE Congress on Evolutionary Computation, IEEE Press, Los Alamitos (2003)
28. Hays, W.: Statistics, 5th edn., USA, Orlando Florida (1994)
29. Cicirello, V., Smith, S.: Wasp-like Agents for Distributed Factory Coordination. Autonomous Agents and Multi-Agents Systems 8, 237–266 (2004)
30. Bussler, C.: Analysis of the Organization Modeling Capability of Workflow-Management-Systems. In: Amoroso, D.L. (ed.) Conference of the Pacific Research Institute for Information Systems and Management (PRIISM '96), Maui, HI, USA (1996)

31. Rupietta, W.: Organizational Models for Cooperative Office Applications. In: Karagiannis, D. (ed.) DEXA 1994. LNCS, vol. 856, Springer, Heidelberg (1994)

32. van der Aalst, W.M.P., Kumar, A., Verbeek, H.M.W.: Organizational Modeling in UML and XML in the context of Workflow Systems. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, Springer, Heidelberg (2004)

33. Momotko, M., Subieta, K.: Dynamic Changes in Workflow Participant Assignment. In: Manolopoulos, Y., Návrat, P. (eds.) ADBIS 2002. LNCS, vol. 2435, Springer, Heidelberg (2002)

34. Shen, M., Tzen, G.-H., Lio, D.-R.: Multi-Criteria Task Assignment in Workflow Management Systems. In: Sprague, R.J. (ed.) 36th HICSS, IEEE, Waikoloa, HI (2003)

35. Huang, Y.-N., Shan, M.-C.: Policy-Based Resource Management. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999. LNCS, vol. 1626, Springer, Heidelberg (1999)

36. Bussler, C., Jablonski, S.: Policy Resolution for Workflow Management. In: Sprague, R.J. (ed.) 28th Hawaii International Conference on Systems Sciences (HICSS 1995), IEEE, Hawaii (1995)

37. Gomoluch, J., Schroeder, M.: Market-Based Resource Allocation for Grid Computing: A Model and Simulation. In: Proc. 1st. Int'l. Workshop on Middleware for Grid Computing (MGC) (2003)

38. Krieger, M.J.B., Billeter, J.-B., et al.: Ant-like task allocation and recruitment in cooperative robots. Nature 406(31), 992–995 (2000)

39. Shehory, O., Kraus, S.: Methods for task allocation via agent coalition formation. Artificial Intelligence Journal 101(1-2), 165–200 (1998)

40. Tan, J.C., Harker, P.T.: Designing Workflow Coordination: Centralized Versus Market-Based Mechanisms. Information Systems Research 10, 328–342 (1999)

41. Alt, R., Klein, S., Kuhn, C.: Service Task Allocation as an Internal Market. In: Baets, W.R.J. (ed.) Second European Conference on Information Systems (ECIS 1994). Nijenrode University Press, Netherlands, pp. 424–432 (1994)

42. Wellman, M.P., Walsh, W.E., Wurman, P.R., MacKie-Mason, J.K.: Auction protocols for decentralized scheduling. Games and Economic Behavior 35, 271–303 (2001)

43. Baggio, G., Wainer, J., Ellis, C.: Applying scheduling techniques to minimize the number of late jobs in workflow systems. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 1396–1403. Springer, Heidelberg (2004)

44. Bonin, G.: Final Report, Assessment of Fire and EMS Services Branchburg Township, New Jersey; Tridata, A Division of Sustem Planning Corporation (2005)

45. Mehrotra, S., Butts, C., Kalashnikov, N.: Project Rescue: Challenges in Responding to the Unexpected. SPIE 5304, 179–192 (2004)

46. Amer, A., Brustoloni, J., Chrysanthis, P.K., Hauskrecht, M., Labrinidis, A., Melhem, R., Mosse, D., Pruhs, K., Comfort, L.: Secure-CITI: A Secure Critical Information Technology Infrastructure for Disaster Management, Hazard Reduction and Response in Metropolitan Regions (2003)

47. Freßmann, A.: Adaptive Workflow Support for Search Processes within Fire Service Organisations, University of Trier (2006)