COORDINATION OF PLANNING AND SCHEDULING TECHNIQUES FOR A DISTRIBUTED, MULTI-LEVEL, MULTI-AGENT SYSTEM

John S. Kinnebrew, Daniel L. C. Mack, Gautam Biswas, Douglas C. Schmidt
Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37203, USA
john.s.kinnebrew@vanderbilt.edu

Keywords: Planning and Scheduling, Agent Cooperation and Coordination.

Abstract: Planning and scheduling for agents operating in heterogeneous, multi-agent environments is governed by the nature of the environment and the interactions between agents. Significant efficiency and capability gains can be attained by employing multiple planning and scheduling mechanisms that are each tailored to the particular agent roles. This paper presents such a framework for a global sensor web, operating as a two-level hierarchy: (1) the mission level for global coordination of complex tasks, and (2) the resource level for operation of subtasks on individual sensor networks. We describe important challenges in coordinating among agents employing two different planning and scheduling methods and develop a coordination solution for this framework. Experimental results validate the benefits of employing guided, context-sensitive coordination of planning and scheduling in such systems.

1 INTRODUCTION

In large-scale, distributed, multi-agent systems (MAS) that span multiple domains of agent operation, choosing a single planning and scheduling mechanism for all agents may be inefficient and impractical. For example, NASA's Earth Science Vision calls for the development of a global sensor web that provides coordinated access to sensor network resources for research and resolution of Earth science issues (Hildebrand et al., 2004). This global sensor web must select and coordinate an appropriate subset of heterogeneous, distributed sensors and computational resources for user tasks that often require collaboration among multiple constituent sensor networks. Complex task execution with resource constraints and time deadlines presents planning, scheduling, and coordination issues at multiple levels of the sensor web.

Our *Multi-agent Architecture for Coordinated Responsive Observations* (MACRO) platform provides a powerful computational infrastructure for deploying, configuring, and operating large sensor webs with many constituent sensor networks (Suri et al., 2007). MACRO is structured as a two-level agent hierarchy: (1) the *mission level* where global coordination across sensor networks is achieved, and (2) the *resource level* where operation of the local sensor network is coordinated and controlled. Agents at these different levels of the system operate in different contexts that imply different planning and scheduling requirements.

At the mission level, multiple sensor networks must coordinate to allocate their limited resources to requested tasks in a manner that provides a high system utility. Moreover, they must coordinate planning and scheduling at an appropriate level of abstraction to avoid computational intractability. At the resource level, individual sensor networks must perform detailed planning and scheduling to complete assigned subtasks in dynamic, uncertain, and resource-constrained, environments.

These differences suggest that significant efficiency and capability gains may be achieved by employing different planning and scheduling techniques tailored to the particular requirements at each level. Developing such an agent architecture, however, also presents challenges in coordinating among the agents that use different planning and scheduling mechanisms. In particular, employing different planning and scheduling mechanisms at the mission and resource levels requires an appropriate translation of the task, plan, and schedule representations between levels. It also requires a coordination mechanism for deciding when to exchange information between levels during plan execution.

The remainder of the paper is organized as follows: Section 2 outlines the key capabilities provided by the MACRO agent framework; Section 3 summarizes the planning and scheduling coordination challenges and the solutions we developed for this paper; Section 4 evaluates experimental results that show the reduction in communication and computation achieved by using MACRO's guided, context-sensitive coordination mechanism for planning and scheduling; Section 5 compares our work with related research; and Section 6 presents concluding remarks.

2 OVERVIEW OF MACRO

To provide global coordination of the sensor web, the MACRO mission level is comprised of *broker agents*, *user agents*, and *mission agents*. Broker agents act as the intelligent system infrastructure, providing matchmaker services, aggregating relevant domain information, tracking system performance, and mediating allocation negotiations (Kinnebrew, 2009). User agents generate the high-level tasks and are typically interfaces to mission scientists and wrappers for legacy systems (*e.g.* weather modeling applications) that can request execution of sensor web tasks. Each mission agent represents an independent *sensor network* and achieves its allocated tasks with the resources available in its sensor network.

As the representative of an entire sensor network, a mission agent straddles the boundary between the mission and resource levels. At the resource level, mission agents divide tasks among the *exec agents*, which are responsible for the operation of the sensor network hardware. Each exec agent controls a set of computational/sensor resources within the sensor network and is supported by additional domain-specific agents. An exec agent also employs services for planning, scheduling, allocation, and resource management of the hardware under its control. These services are shared with any supporting agents under its direction, providing a centralized control and environmental awareness for its set of resources.

2.1 MACRO Mission Level

At the mission level of a sensor web MAS, user tasks and scheduled plans spanning multiple sensor networks have a high degree of complexity. Hierarchical analysis helps deal with this complexity, both for problem/task representation by domain experts and for coordinated planning and scheduling among multiple agents. Therefore, MACRO employs a modified implementation of the *Task Analysis, Environment Modeling, and Simulation* (TÆMS) (Horling et al., 1999) language, which provides a hierarchical task network representation for multi-agent planning and scheduling.

MACRO incorporates the OGC SensorML (Botts et al., 2007) representation of sensors and data processing with the TÆMS hierarchically decomposable task representation. This provides standardized descriptions of task/subtask requirements and effects across sensor networks. The TÆMS representation also allows the specification of discrete probability distributions for task/subtask characteristics including potential outcome quality and duration (Lesser et al., 2004).

To coordinate and schedule TÆMS tasks across sensor networks, MACRO mission agents employ the *Generalized Partial Global Planning* (GPGP) (Lesser et al., 2004) coordination mechanism, which works in conjunction with a planning and scheduling mechanism that can generate an appropriate task decomposition and schedule from a TÆMS task tree. For this purpose, MACRO mission agents employ *Design-To-Criteria* (DTC) (Wagner and Lesser, 2001) planning/scheduling, which has successfully been used in conjunction with GPGP coordination (Lesser et al., 2004). DTC scheduling is a soft real-time, heuristic approach to solving the combinatorial problem of optimally decomposing and scheduling a TÆMS task. DTC is particularly suited to the MACRO mission-level because it can optimize plans and schedules based on user-provided criteria, such as minimizing execution time or maximizing expected quality.

2.2 MACRO Resource Level

Exec agents use the *Spreading Activation Partial Order Planner* (SA-POP) (Kinnebrew et al., 2007), which generates high utility, scheduled, partial order plans that respect local resource constraints. SA-POP allows the exec agents to use their limited computational resources to maximize expected utility for achieving local goals in the dynamic, uncertain environments common to the resource level. Moreover, SA-POP provides incremental re-planning/re-scheduling that can quickly revise scheduled plans during execution and prevent more expensive re-planning/re-scheduling at the mission level. In conjunction with SA-POP, exec agents also employ the *Resource Allocation and Control Engine* (RACE) (Shankaran et al., 2007) for resource allo-

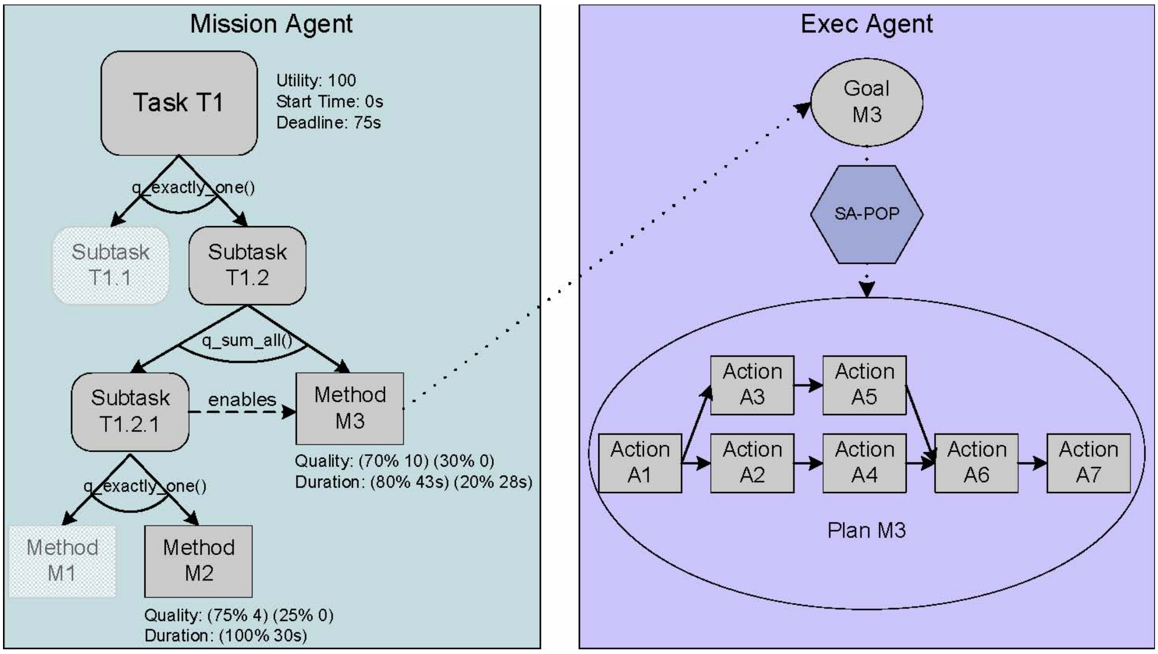

Figure 1: Planning/Scheduling Representations in MACRO

cation and management to meet scheduled deadlines and required quality of service (QoS) parameters for deployed applications and hardware-based actions.

First-principles planning and scheduling with SA-POP requires a set of goal conditions that correspond to the desired outcome. These goal conditions are specified as desired environmental and system conditions with associated utility values and time deadlines. Given these goal conditions, SA-POP uses current/-expected conditions to generate a scheduled plan of high expected utility (Kinnebrew et al., 2007).

3 MACRO COORDINATION

As described in Section 2.1, mission agents must efficiently generate and coordinate plans and schedules provided by the TÆMS task decomposition trees and criteria-directed scheduling. As shown in Figure 1, the leaves of a TÆMS task tree are *methods*, which in standard TÆMS usage can be directly executed by the agent. In MACRO, however, mission agents must communicate these methods to their exec agents for resource-level planning/scheduling and actual execution. At the resource level, the decision-theoretic, first-principles planning and constraint-propagation scheduling is efficiently performed by SA-POP for achievement of *goals* in the dynamic sensor network environment shown in Figure 1. Effectively employing both representations and forms of planning and scheduling presents multiple challenges for coordination between MACRO mission and exec agents.

3.1 Translation: Top-Down

Problem. For an exec agent to implement a TÆMS method, the mission agent must translate it into the goal format used by SA-POP. SA-POP goals include one or more goal conditions with associated utility values and time deadlines. To plan for a goal accurately, SA-POP requires knowledge of expected system and environmental conditions at the time the plan will be executed. Although current conditions and other exec agent plans provide most of this information, other expected conditions may be the result of methods assigned to other exec agents in the mission agent's current plan (*i.e.*, other methods that *enable* the method in question by satisfying some of its preconditions).

Solution → Cross-references in task/goal modeling. In MACRO, domain experts (*e.g.*, scientists and engineers who design and deploy the sensor network) employ a domain-specific modeling language, defined using the Generic Modeling Environment (GME) (Karsai et al., 2003)), to specify the TÆMS task tree for a mission agent. In this model, TÆMS methods are associated with necessary resource-level preconditions and goal conditions, which are, in turn, represented in the action network model employed by

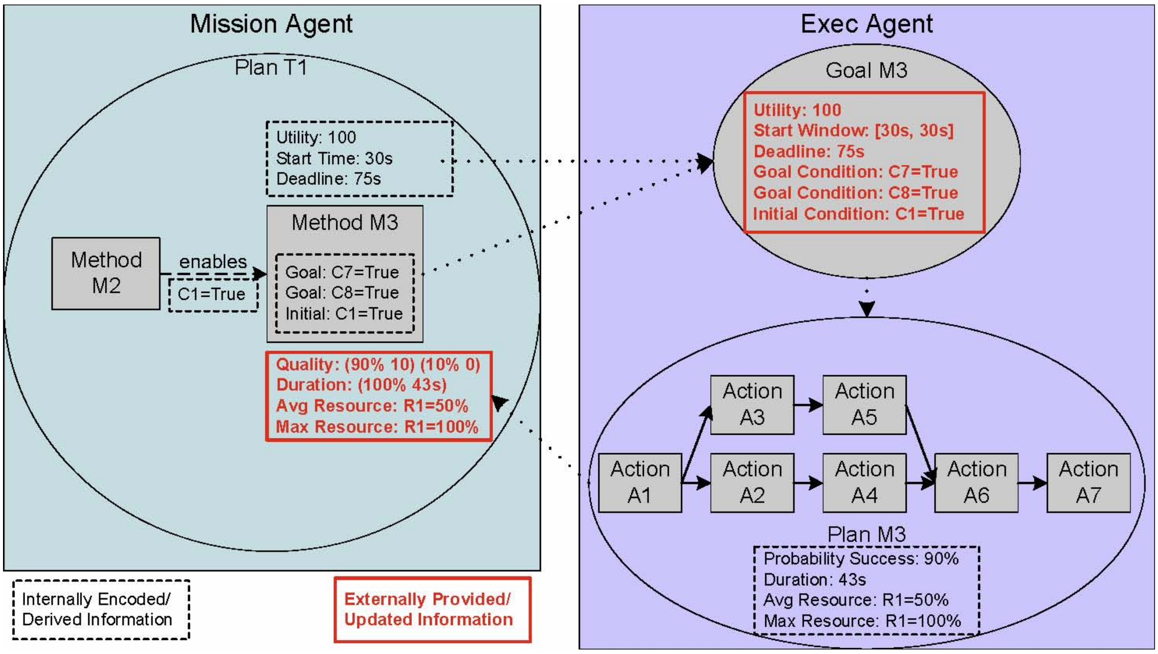

Figure 2: Planning/Scheduling Translation in MACRO

the exec agent and SA-POP. Moreover, the domain expert can automatically derive method distributions for duration and outcome in this model by providing potential initial condition settings (with an associated probability) to SA-POP, which produces scheduled plans and summarizes their probability of success, expected duration, and resource usage.

Instead of directly executing a method, the mission agent uses the encoded translation information from the model to provide a goal to the exec agent. This top-down translation is shown by the mission agent to exec agent information transfer in Figure 2. The mission agent awards overall task utility to methods based on the quality aggregation functions (QAFs) and expected quality in the TÆMS task tree.

In the chosen decomposition of the TÆMS task tree, parents with a QAF that requires execution of all child subtasks/methods pass the full parent utility to each child, while QAFs that allow any subset of children pass a percentage of parent utility based on the child's percentage of total expected quality for the parent. For example, a task with an overall utility of 100 that is decomposed into two subtasks of expected quality 3 and 7 with a sum QAF would assign utility of 30 and 70, respectively, to its subtasks. In future work, we intend to investigate more advanced methods of reward assignment in the decomposition of TÆMS task trees.

3.2 Translation: Bottom-Up

Problem. Another important challenge is codifying the bottom-up translation between SA-POP plans and TÆMS method parameters. Standard TÆMS methods include a priori probability distributions for duration and outcome quality, which are used during initial criteria-directed scheduling by the mission agent. After an exec agent plans to achieve a goal, the resultant scheduled plan may imply significantly different probability distributions for the corresponding method. Similarly, as a plan is being executed by the exec agent, there may be further changes to the expected duration or probability of outcomes for the plan and its corresponding method. To improve the efficiency of future criteria-directed scheduling and to trigger appropriate mission-level re-scheduling, information about the exec agent's plan must be communicated to the mission agent.

Solution → Summarize resource-level plans. Instead of providing the complete resource-level plan to the mission agent (whose format is ill-suited to its planning and scheduling capabilities), a MACRO exec agent summarizes its plan by providing relevant information only, including (1) expected duration, (2) probability of achieving the goal, and (3) average and maximum resource usage over expected execution. The mission agent uses these values to update method parameters with more accurate information, based on the resource-level planning and scheduling for the current and expected environmental/system conditions. The updated method parameters allow the mission agent to more effectively perform any further planning and scheduling for its task(s).

3.3 Context-Sensitive Updates

Problem. In addition to translating between the mission and exec agent planning/scheduling representations, MACRO agents must also decide *when* to update and communicate the translated information. In particular, during execution of exec agent plans, deviations may occur (*e.g.*, differences between actual and expected duration of actions). Only some variations, however, will impact the rest of the mission-level plan—or other plans—in a manner that would be of interest to the mission agent.

Solution → Leverage mission-level task context. Given the hierarchical relationship between mission and exec agents, the top-down decision to communicate (*i.e.*, when the mission agent should communicate information to an exec agent) is relatively straightforward. Specifically, whenever a new task is decomposed/scheduled or method parameters in the plan are changed by re-planning/re-scheduling, the mission agent communicates the new or revised goals (translated from the methods) to the assigned exec agents.

For bottom-up updates, however, an exec agent can use its knowledge of a mission agent's overall goals/interests to guide its decision of when to communicate. Without mission agent guidance, an exec agent would be forced to communicate on a periodic basis or whenever the execution deviates from the scheduled plan, which may happen frequently in a dynamic sensor network environment. When tasking an exec agent with a goal, therefore, the MACRO mission agents also provide guidance and contextual information, such as the optimization criteria for the related task. Knowledge of the optimization criteria allows the exec agent to configure SA-POP's planning and scheduling to prefer plans based on that criteria.

In addition to optimization criteria, the mission agent can specify thresholds for deviation (in either direction) of an executing plan on success probability, expected utility, duration, and resource usage. This information provides the exec agent with guidance on the *context* for the corresponding method in the mission agent's plan. This context allows the agent to more intelligently determine when to update its scheduled plan and provide the revised summary to the mission agent. Specifically, during execution of a plan, the exec agent will only re-plan and re-schedule if the expected utility falls below an under-threshold or if the duration surpasses an over-threshold. Whenever any other threshold is exceeded, the exec agent will simply communicate updated summary information to the mission agent.

Figure 3 shows the execution of the resource-level plan from Section 3.2. To demonstrate the benefit of

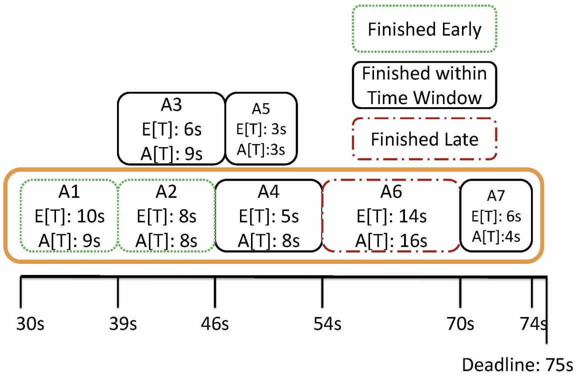

Figure 3: A Resource-level Plan (Critical Path Highlighted)

the guidance/context provided by the mission agent, we focus on deviations of action duration from expected duration in the critical path (*i.e.*, the linked sequence of actions that requires the longest time to complete). Although the planning and scheduling in MACRO does not rely on identification of the critical path, such a path(s) always exist, and it constrains the expected completion time of the plan.

Without the context provided by an over-threshold on duration, the exec agent would have no knowledge of what deviations were important to the mission agent and would have to communicate updates based on each deviation. It would recalculate its schedule every time an action did not complete with exactly its expected duration. Further, it would have to transmit the new expected duration of the plan either with every recalculation, or at least every time an action finished outside of its scheduled end window (either before or after that window).

The example execution in Figure 3 shows a typical case in which the mission agent provides an over-threshold on duration equal to the difference between the expected end-time of the plan and the original deadline. In other words, the mission agent is only interested in changes to the resource-level schedule that would result in its finishing later than the deadline. In this example, the exec agent would have to re-plan/re-schedule only when execution of action A6 goes beyond its scheduled end window. Without the appropriate context, in the form of the duration threshold, the exec agent would have also had to unnecessarily recalculate or re-plan/re-schedule three times (after completion of A1, A4, and A3) and communicate unnecessary updates twice (after A1 and A4).

4 COORDINATION RESULTS

This section presents the results of mission and exec agent coordination through the simulated execution of randomly-generated resource-level plans with a variety of duration distributions for actions. These results validate our claims in Section 3 that MACRO's use of guided, context-sensitive coordination in planning/-scheduling can reduce communication and computation, while still providing relevant information in a timely fashion.

4.1 Experimental Design

Our experiments simulate a scheduled, partial-order plan generated by SA-POP for an exec agent at the resource level of MACRO. These plans include a set of actions with expected start and end time windows, as well as ordering links. For these experiments, we only simulate cases in which a valid plan can be generated.

One experimental parameter is the variability of actual durations for actions, which requires different probability distributions parameterized by a sigma value. The experiments included both uniform distributions and Gaussian (Normal) distributions, although the results for the uniform distributions are omitted due to length constraints. The uniform distributions, however, showed the same trends observed in the Gaussian distributions and required even less MACRO computation and communication for both the baseline and context-sensitive coordination mechanisms. The action duration distributions have a mean of 100 seconds and "low" and "high" variance scenarios providing a 95% likelihood (for the Gaussian) that durations are within 25 seconds or 75 seconds of the mean, respectively.

Another experimental variable is the length of the critical path. The distributions provide all actions with an expected duration of 100 seconds. The expected time for completion of the plan therefore depends solely on the number of actions in the critical path.

The final experimental variable is the time threshold provided by the mission agent, which determines how far actions can surpass their expected end times before re-planning and re-scheduling are required for MACRO context-sensitive coordination. In these experiments, we used a worst case scenario in which neither the resource-level nor mission-level plans could be changed during execution. Whenever an action execution went beyond its scheduled end window, therefore, the schedule was updated and communicated to the mission agent but no changes to the plan or threshold were made. Disallowing re-planning/re-scheduling allows the experiments to use randomly-generated plans across a range of parameters rather than a few example problems. It also results in significantly more computation and communication, however, since future actions are likely to continue going beyond their end windows after a critical path action's end window is exceeded.

4.2 Experimental Results

Each experimental run included 10,000 trials with the given parameter settings. In each trial, a series of (n) actions formed the critical path, and each action had an expected duration of 100 seconds. Using the chosen distribution, random values are generated that correspond to actual execution times. The number of updates and messages are calculated using those values.

4.2.1 Investigating Critical Path Length

These experiments were performed under the assumption that the mission agent simply requires a method to complete by the provided deadline and should only be notified if the expected execution will exceed that deadline. The threshold value is therefore set to the difference between the deadline and the expected duration of the plan. This threshold is varied in the experiments between 0 seconds and 200 seconds by 5 second increments.

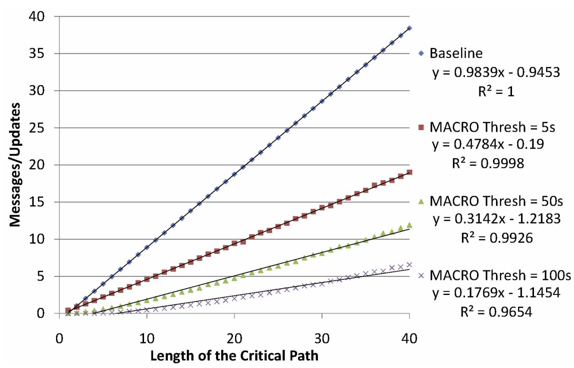

Figure 4: The effect of critical path length with selected thresholds with a low variance Gaussian

Figure 4 shows how MACRO's use of threshold information from the mission agent results in significantly less computation and communication than the baseline for everything but the smallest of critical paths The linear nature of the data suggests that

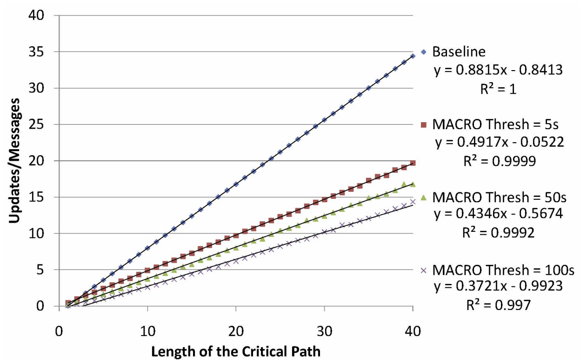

Figure 5: The effect of critical path length with selected thresholds with a high variance Gaussian

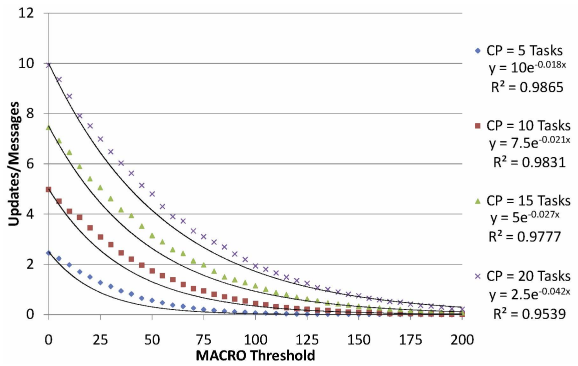

Figure 6: The effect of Slack with selected critical paths on a low variance Gaussian

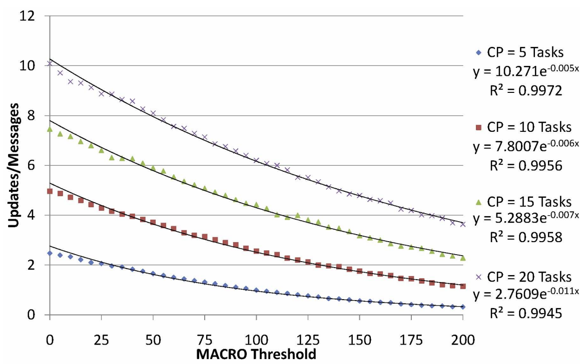

Figure 7: The effect of Slack with selected critical paths on a high variance Gaussian

in the worst case (*i.e.*, a tight threshold/deadline), MACRO sends about half as many messages as the baseline. As the threshold increases, MACRO performs even better, whereas the baseline performance does not change.

A comparison of the low variance action duration distribution in Figure 4 to a high variance one in Figure 5 shows that with the smallest thresholds a ratio of approximately 1 update per 2 actions in the critical path is required for both distributions. The 1:2 ratio is thus an approximate upper limit on the average number of updates required in MACRO, even when re-planning and re-scheduling is not possible.

The baseline mechanism shows a slight, relative improvement in the high variance case, but MACRO's context-sensitive coordination still requires far fewer updates. However, the number of updates required in MACRO with different thresholds are much closer in the high variance case than the low variance case. This result suggests that when action durations are less certain, the critical path length is significantly more important than the threshold, because even large thresholds can be exceeded by a series of actions that begins with an unexpectedly long-running action.

4.2.2 Investigating Time Thresholds

Figure 6 and Figure 7 show the trends in communication and computation with respect to the duration threshold. The baseline is not included in these figures because it does not make use of the threshold value. If included, it would be a constant line close to the number of actions in the critical path.

The trend in these results shows that as the threshold increases, the number of MACRO updates decreases close to exponentially. This exponential decay results from the fact that longer thresholds allow a series of actions to exceed their expected duration by a greater amount before requiring an update, but that extreme variation from expected durations can occur and will still require some updates, even with relatively large thresholds. These results also show, however, that even when uncertainty of action duration is high, the exec agent can leverage the contextual information provided by the mission agent to minimize computation and communication.

5 RELATED WORK

MACRO's approach to planning and scheduling builds upon and extends a significant body of related work. At the mission level, MACRO agents employ *Design-To-Criteria* (DTC) planning/scheduling (Wagner and Lesser, 2001) operating on an augmented TÆMS task tree to efficiently optimize for relevant criteria in generating a scheduled plan to perform assigned subtasks. While, at the resource level, exec agents employ SA-POP (Kinnebrew et al., 2007) for decision-theoretic planning with constraint-propagation scheduling.

MACRO coordinates agents employing its two planning/scheduling mechanisms to communicate the most useful information at an appropriate abstraction level and at the right time. The translation from resource-level plans to mission-level method parameters has some similarities to research that uses plan summary information to coordinate between agents employing HTN planning (*e.g.*, (Clement and Durfee, 1999; Clement and Durfee, 2000)). MACRO mission and exec agents, however, employ *different* representations for planning and scheduling. Moreover, the resource and scheduling constraints in MACRO require summary information beyond the pre-, in-, and post-conditions used in Clement's task summary info approach (Clement and Durfee, 1999).

The MACRO translation between mission agent methods and exec agent goals provides similar planning summary information at a level of abstraction determined by the hierarchical divide between their domain representations. In addition, MACRO exec agents summarize scheduling, probability, and resource-usage information that can be used by mission agents employing TÆMS task tree decomposition with criteria-directed scheduling.

6 CONCLUDING REMARKS

This paper presents some key research challenges for coordinating planning and scheduling at two levels of a hierarchical multi-agent system. In particular, we discuss MACRO's solutions to coordinating HTN task decomposition with criteria-directed scheduling and first-principles decision-theoretic planning with constraint-propagation scheduling. Finally, we conducted experiments that showcased the benefits gained by employing MACRO's guided, context-sensitive coordination of planning and scheduling.

Our experimental results quantified the effects of different distributions from which average duration information is derived for resource-level actions. The experiments also showcase the effects of other planning/scheduling parameters, including the length of a scheduled plan's critical path and the restrictiveness of the deadline. Moreover, our results verify the scalability of MACRO planning/scheduling coordination when execution time is the primary criteria of interest to the mission agent. In future work, we intend to explore other forms of utility assignment in TÆMS task tree decomposition and evaluate the benefits of context-sensitive coordination with thresholds on plan characteristics other than execution time.

REFERENCES

Botts, M. et al. (2007). Sensor Model Language (SensorML). Technical Report OpenGIS Implementation Specification Document 07-000, Open Geospatial Consortium.

Clement, B. and Durfee, E. (1999). Theory for Coordinating Concurrent Hierarchical Planning Agents using Summary Information. *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Innovative Applications of AI Conference*, pages 495–502.

Clement, B. and Durfee, E. (2000). Performance of Coordinating Concurrent Hierarchical Planning Agents using Summary Information. *Proceedings of the Fourth International Conference on Multi-Agent Systems*, pages 373–374.

Hildebrand, P., Wiscombe, W., Albjerg, M., Booth, J., Miller, R., Miller, T., Mlynczak, M., Paules, G., Peterson, D., Raymond, C., et al. (2004). NASA Earth Science Vision 2030: Working Group Report. Technical Report NP-2003-2-611-GSFC, NASA.

Horling, B., Lesser, V., Vincent, R., Wagner, T., Raja, A., Zhang, S., Decker, K., and Garvey, A. (1999). The TAEMS White Paper. Technical report, Multi-Agent Systems Lab, University of Massachusetts.

Karsai, G., Sztipanovits, J., Ledeczi, A., and Bapty, T. (2003). Model-Integrated Development of Embedded Software. *Proceedings of the IEEE*, 91(1):145–164.

Kinnebrew, J. S. (2009). Global Sensor Web Coordination and Control in a Multi-agent System. In *Proceedings of the AAAI Doctoral Consortium (at IJCAI '09)*.

Kinnebrew, J. S., Gupta, A., Shankaran, N., Biswas, G., and Schmidt, D. C. (2007). Decision-Theoretic Planner with Dynamic Component Reconfiguration for Distributed Real-time Applications. In *Proceedings of the 8th International Symposium on Autonomous Decentralized Systems (ISADS 2007)*, Sedona, Arizona.

Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., Prasad, M., Raja, A., et al. (2004). Evolution of the GPGP/TÆMS Domain-Independent Coordination Framework. *Autonomous Agents and Multi-Agent Systems*, 9(1):87–143.

Shankaran, N., Schmidt, D. C., Chen, Y., Koutsoukous, X., and Lu, C. (2007). The Design and Performance of Configurable Component Middleware for End-to-End Adaptation of Distributed Real-time Embedded Systems. In *Proc. of the 10th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC 2007)*, Santorini Island, Greece.

Suri, D., Howell, A., Schmidt, D. C., Biswas, G., Kinnebrew, J., Otte, W., and Shankaran, N. (2007). A Multi-agent Architecture for Smart Sensing in the NASA Sensor Web. In *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, Montana.

Wagner, T. and Lesser, V. (2001). Design-to-Criteria Scheduling: Real-Time Agent Control. *Lecture Notes in Computer Science*, pages 128–143.