Design-to-Criteria Scheduling:
Real-Time Agent Control *

Thomas Wagner¹ and Victor Lesser²

¹ Computer Science Department
University of Maine
Orono, ME 04469
wagner@umcs.maine.edu

² Computer Science Department
University of Massachusetts
Amherst, MA 01003
lesser@cs.umass.edu

Abstract. Design-to-Criteria builds custom schedules for agents that meet hard temporal constraints, hard resource constraints, and soft constraints stemming from soft task interactions or soft commitments made with other agents. Design-to-Criteria is designed specifically for online application – it copes with exponential combinatorics to produce these custom schedules in a resource bounded fashion. This enables agents to respond to changes in problem solving or the environment as they arise.

1 Introduction

Complex autonomous agents operating in open, dynamic environments must be able to address deadlines and resource limitations in their problem solving. This is partly due to characteristics of the environment, and partly due to the complexity of the applications typically handled by software agents in our research. In open environments, requests for service can arrive at the local agent at any time, thus making it difficult to fully plan or predict the agent’s future workload. In dynamic environments, assumptions made when planning may change, or unpredicted failures may occur¹. In most real applications,

¹ This differs from states that are explicitly recognized and planned for [1] as software agents may be required to perform a different set of tasks, as well as having to react to changes in the environment.
deadlines or other time constraints are present on the agent’s problem solving [16, 8]. For example, in an anti-submarine warfare information gathering application [3], there is a deadline by which the mission planners require the information. Resource limitations may also stem from agents having multiple different tasks to perform and having bounded resources in which to perform them. Temporal constraints may also originate with agent interactions – in general, in order for agent $\beta$ to coordinate with agent $\alpha$, the agents require mutual temporal information so that they can plan downstream from the interaction.

In this paper, we focus on the issue of resource bounded agent control. We use the term resource bounded to denote the existence of deadlines and of other constraints like cost limitations or application specific resource limitations (e.g., limited network bandwidth). Where it is important to differentiate hard and soft deadlines from these other constraints, we refer to them explicitly.

For agents to adapt rationally to their changing problem solving context, which includes changes in the environment and changes to the set of duties for the agent to perform, they must be able to:

1. Represent or model the time and resource constraints of the situation and how such constraints impact their problem solving. We believe this must be done in a quantified fashion as different constraints have different degrees of effect on problem solving.
2. Plan explicitly to address the resource limitations. In our work, this may imply performing a different set of tasks, using alternate solution methods, or trading-off different resources (or quality), depending on what is available.
3. Perform this planning online – in the general case, this implies coping with exponential combinatorics online in soft real time.

While the first two requirements obviously follow from the domain, the third requirement is less obvious. Agents must be able to perform real time control problem solving online because of the dynamics of the environment. If it is difficult to predict the future and there is a possibility of failure, or new tasks arriving, agents will, by necessity, have to react to new information and replan online.

The Design-to-Criteria (DTC) agent scheduler and the TÆMS task modeling framework are our tools for addressing these requirements and achieving resource-bounded agent control (Figure 1). TÆMS provides agents with the framework to represent and reason about their problem solving process from a quantified perspective, including

---

2 Including resources uncontrollably becoming more or less constrained. For example, network latency increasing due to some activity other than the agent’s problem solving.