

Interfaces for Team Coordination *†

Romeo Sanchez, Jing Jin, Rajiv T. Maheswaran and Pedro Szekely
Information Sciences Institute, University of Southern California
4676 Admiralty Way Suite 1001, Marina del Rey, CA 90292
{rsanchez, jing, maheswar, pszekely}@isi.edu

ABSTRACT

Coordinators are intelligent software agents that help humans to collaborate and coordinate execution of multiple activities in dynamic, distributed and uncertain domains. One of the main challenges is the generation of effective user interfaces for team coordination. Intelligent interfaces that avoid information overload, and facilitate user interactions to repair mission plans are needed, and they are the focus of this work.

INTRODUCTION

Our work focuses on distributed teams of humans who must coordinate to achieve a common goal. Consider teams of fire-fighters fighting a wild fire, rescue workers working on relief operations after a natural disaster, or units in military operations. The team starts its mission with a plan specifying who should do what and when. Due to the dynamics of these environments, activities may fail or take longer than expected, or new tasks are discovered. In consequence, team members must coordinate to repair the initial plan.

To assist with coordination, each team member (unit) is equipped with a portable computer. An agent called the *Coordinator*, running on each computer, keeps track of the activities that the unit should perform, pre-planned contingencies, and of the dependencies with other units. Each agent has a partial model of the tasks that the whole team must accomplish, so agents must coordinate to compute plan repairs as the mission unfolds. They may need to adjust the timings of their activities, or they may need to adopt contingency plans to achieve sub-tasks in a different way or to address new tasks.

*The work presented here is funded by the DARPA COORDINATORS Program under contract FA8750-05-C-0032. The U.S. Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them. Approved for Public Release, Distribution Unlimited.

†Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. IUI'08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain. Copyright 2008 ACM 978-1-59593-987-6/08/0001 \$5.00

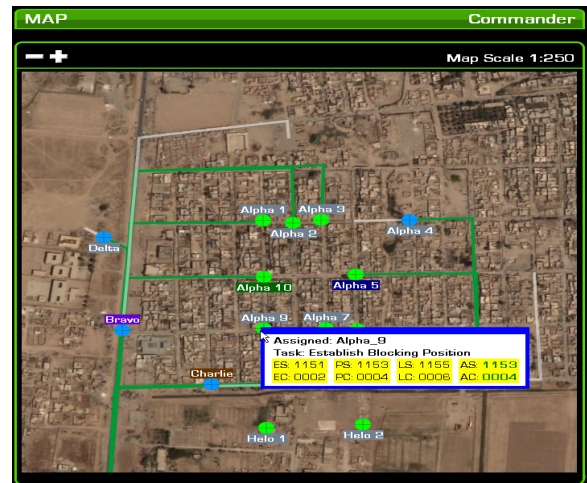


Figure 1: Map View

This paper focuses on the user interfaces to help the units stay coordinated. The interfaces show users the set of activities that they should perform, provide status about dependent units, provide information about imminent coordination problems and help users evaluate and choose plan repair options that the agents compute.

THE COORDINATORS USER INTERFACE

The key challenge in the design of the user interface is to avoid information overload, to show users only the information they need to stay coordinated, and to streamline the interaction when users need to repair the plan. The focus of the users is not the software, but the tasks they are engaged in. There are two types of users in the application, a regular coordinator (unit) and a *Commander*. While the Commander's interface shows information about all units in the mission, an individual unit's interface shows only data about dependent units, those that the current unit depends on to carry out its activities, or related units that will be affected by the unit's actions. The Coordinators user interface consists of three views: (1) a map, (2) a task, and (3) a schedule view.

The map view, which can be seen on Figure 1, shows the locations of units, the paths they have followed or are about to follow, and their status information. The status of the units is shown as executing an activity (blue), idle/waiting (gray), activity completed successfully (green), or unit requires attention (yellow). A pop-up box shows detailed information about each unit's current activity.

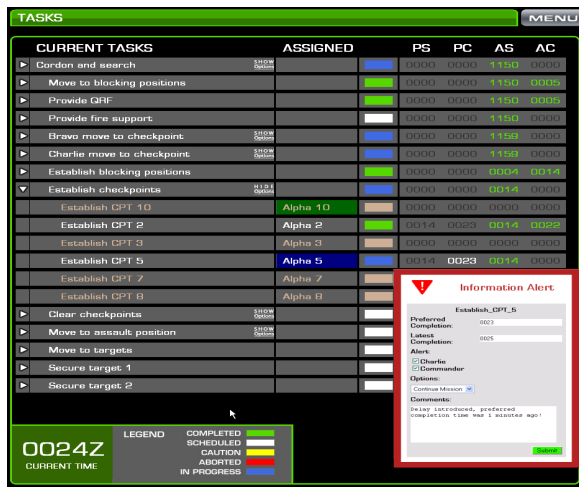


Figure 2: Task View

The task view on Figure 2 shows a hierarchical view of the tasks that a unit is engaged in, and also includes dependent tasks from other units. In the case of a commander, it will show the hierarchical decomposition of the whole plan allowing them to drill down to show arbitrary levels of detail. This view is mainly of interest to the commander who is concerned with the overall operation. Individual units may choose to switch to this view when they run into trouble and need to engage contingency plans. This view also shows the timing information for all activities focusing on earliest, preferred, latest start and end times; and the progress that has been made towards the main tasks in the mission.

Figure 3 shows the schedule view of the plan. This is the main screen for the units as it shows a Gantt chart of the scheduled activities that the unit should perform, and other dependent activities. A vertical red line shows the current time, and gives each unit a good overview of how it is progressing according to the current plan. It clearly shows the status of the activities in the plan by marking them as executing (blue), waiting (white), aborted (red), and completed (green). Indicating also when an activity may lead to a potential problem with a yellow bar. For the commander, this view will also show a set of available options when a problem is detected, and the effects of those options in the current schedule.

The user interface is very interactive. When a unit fails to complete an activity at its preferred time, the Coordinator will alert the unit of the impending delay. The alert dialog box (seen on Figure 2) features easy to select options to enable the unit to specify that it will complete by the latest completion time (i.e., in time), that it will be delayed by a specific amount of time, or that it will not be able to complete the activity and recommends aborting it. The coordinator even shows the set of affected units to whom to coordinate to resolve the issue. The agent uses this information to coordinate with other agents to revise the plan. If the plan can be revised without significant changes to the schedules and activities of other units, the revisions will be directly communicated to related agents. If more drastic re-

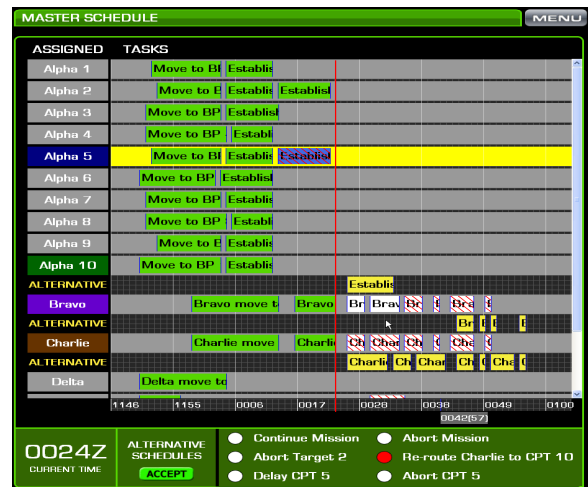


Figure 3: Schedule View

visions are needed, the unit is marked as being in trouble, and its icon becomes yellow on all Coordinators showing it. In addition, the commander will be alerted. The commander can ask the Coordinators for options to repair the plan.

The Coordinators use distributed planning and scheduling techniques to compute options, and as mentioned before these are shown to the commander in the schedule view. Figure 3 shows a situation where one unit (Alpha 5) has ran into trouble. The interface shows the commander alternative courses of action at the bottom of the screen. The commander can select an alternative and compare it to the current plan to see which units and activities will be affected. When the commander chooses an option, he or she can send a coordination alert to affected units to ask them whether they can execute the selected option. The Coordinators compute the options based on the assumptions in the model, and the model can be incorrect, so it is prudent to ask the affected units. The commander can see the responses as they come in, and can at any time engage an option and distribute the schedule changes to affected units. Finally, the interface of the affected units will show the updated plan.

DEMO DESCRIPTION

The Coordinator user interfaces will be shown in the context of a military scenario. The demonstration will show how the interfaces enable all users to stay informed during plan execution, and how the system helps users repair a plan that was derailed due to failure in one activity. The demonstration highlights the effectiveness of the agents and the user interface in intelligently alerting only the affected units, and on enabling the commander to easily and quickly evaluate options to repair the broken plan. The Coordinator user interfaces are deployed using an AJAX Web-based architecture. The browser-based user interface updates automatically by asynchronously retrieving data from the server. Our interfaces for team coordination leverage the advantages of AJAX, namely, non-intrusive, cross platform, easy to deploy, distributed and easy to maintain. A movie of the demo is at <http://coordination.isi.edu/home/cc-demo/cc.html>.