CS 396 Special Topics in Artificial Intelligence
Project Deliverable 2

Ari Wilson

February 15th, 2007

Accomplishments since the last deliverable (within the context of the TeamBots simulation environment) include constructing an empty environment, defining the number and type of resources to be collected, and placing obstacles. After some smaller scale trials are completed, I plan on eventually simulating around 8-10 agents with 2-3 primary resource wants and 2-3 secondary wants, with about 15 different resource types overall. More may be added if I can automate parts of my analysis and I have enough compute time. I have not yet finished constructing a reference generic empty agent with move and communicate primitives as was my intention by this point in the project. My other intention by this point was to extend the RoboComm package bundled with TeamBots that allows for asynchronous simulated agent communication but I have not yet clarified the exact format and parameters of all the messages that will be sent by agents within the system. Of course, the messages that will be sent from one agent to another include placing a bid for an item with certain parameters, starting an auction by advertising an item, communicating about coordinating strategies, and inquiring about items among others.

I am behind the schedule because I thought it would be easier to get started with the TeamBots package than it turned out to be and because I allocated too much work for the first week of the project without realizing how close the deadline was. However, I plan to complete the items not finished for the current deliverable and be on schedule with the next deliverable. At this juncture, it is too early to determine what aspects of this project will change from the project summary submitted only a week ago.

As far as multiple agent communication relates to my project, there will be a need for agents with different strategies to communicate to bid on items, indicate their possession of various items. In KQML this would mostly be done through the use of the tell performative, to place a bid with certain parameters, start an auction or communicate about coordinating strategies, or the ask-if or ask-all performative, to inquire if particular agents have an item that another agent needs. It may be the case that an agent's strategy may involve lying to other agents, so the deny performative may be useful as well. However, most specific information and strategizing will take place in the domain-specific representation language I develop for this project. All agents have identical capabilities and are operating in a known environment with strict parameters and a known goal so many of the other standard performatives are superfluous. All agents will able to communicate to every other agent in an understandable format as they are all created by myself, so some of the interoperability benefits are lost for KQML. These same statements holds for FIPA as well. It may turn out that a language like KIF (Knowledge Interchange Format) may prove useful to implement within the context of the actions that will be performed in this project.

On the other hand, implementing a language like KQML or FIPA within my project will not teach me anything instructive about how to build high-performance learning economic agents. I will have to recast notions of strategy and economic coordination and competition within such

a framework. Including KQML or FIPA will also ensure that I spend a fair amount of time implementing a subset of the chosen ACL and ensuring that I am in compliance with the standards, instead of focusing on the problems that interest me. As it stands, the implementation effort of making my agents communicate via such an ACL does not seem to balance with the benefits it might bring.

Finally, I note that the missing details in the previous project deliverable have been added to the document that went with it and that file has been re-uploaded to the project web page.