Approaching negotiation automation for software development companies

Boris Kötting
University of Kaiserslautern
AG Knowledge Based Systems
P.O. Box 3049
67653 Kaiserslautern, Germany
koetting@informatik.uni-kl.de

Frank Maurer
University of Calgary
2500 University Dr NW
Calgary, Alberta, T2N 1N4 Canada
maurer@cpsc.ucalgary.ca

ABSTRACT
Globally operating software development companies can achieve a better performance when they can distribute tasks within a company efficiently and optimise the allocation of tasks to people with the right skill sets. Doing this manually usually results in a none-optimal assignment of team members to tasks because the local manager often does not know the people with the right skill sets. Negotiating software development tasks is a complex operation, also within a company. You need to make decisions about many different aspects of what information to provide to other employees and how you want to negotiate with them. This abstract deals with our approach to negotiation automation for agents within a globally operating company.

We created a virtual marketplace that supports contract negotiation between different companies as well as within a company. Agents can negotiate on simple or composite tasks by defining the objects of interest and making offers and bids by varying several dimensions of the object. In our marketplace system it is possible to use different negotiation strategies:

First, a simple Yes/No Offer, where a bidder can accept or leave an offer. Second, a sealed bid auction, where every actor can place one bid on the marketplace. Third, an English auction, where the actor's bid must be *better* than the actual bid. We called the forth and most flexible strategy *negotiation*. The offering agent can define which parts of the offer are negotiable. Counterproposals are possible in this approach, so negotiation chains are possible.

There are two interesting scenarios for our first approach of automating negotiation, both concerning the distribution of work within a company.

1) An employee can start her bidding agent, defining which kind of tasks she is willing to do. Her automated agent tries to negotiate with other agents to achieve tasks for her automatically. She will be notified when the agent is successful and starts working on the new task, depending on its schedule.

2) The manager has profiles of all available employees. An automatic negotiation agent will be started automatically for every available employee. The agent is pre-configured for bids for task matching his profile. In this case, the manager can simply place the tasks on the marketplace without assigning them to specific agents, saving a lot of time. Nevertheless, she can send tasks to specific agents by addressing them directly.

There are different extensions necessary for supporting automatic negotiations. We evaluated similarity approaches and utility functions and decided to use an utility function with a specific threshold. The utility function depends on the personal settings of each user. A user can define which parts of offers are interesting for him and can define values or functions for these parts. He also can weight the selected parts of interest differently. As an example, he can say that it is most important to get tasks, which require the skill of a programmer, so he weights it with 50%. The payment and the right schedule are equally behind and so weighted with 25% each. The utility function values every offer depending on specific user settings and sums up the values for the different part to one value. Parts of the offer that are not mentioned by the user are ignored by the utility function. There are two interesting thresholds: a lower threshold for the beginning of the interesting valuation area and a higher threshold for a valuation that is acceptable. When the value of the utility function is higher then the lower threshold, the aim is to get a better offer, so the higher threshold will be reached. At first, the agent will check the negotiation type. For a Yes/No Offer it will ignore the task because it can just accept the offer that is acceptable. For an English auction, it will also ignore the task, because the utility function for an English auction will be only worse (the payment will be lowered in an English auction, which usually also lowers the utility function, except payment is irrelevant for the agent, which seems not realistic). For a sealed auction the agent will create an offer that is tops the higher threshold and the same will be done in case

of negotiation as chosen negotiation type. We currently work on strategies *how* to create this bid. When a task is higher than the second threshold, the agent will make a bid for the task, independently of the negotiation type. An offer to an English auction will be continued until the actual offer for the task is below the second threshold.

A major point to mention is the usability of the user interfaces: the user should get an understanding on the aspects he is working on, but on the other hand details like utility functions should be hidden from him.

Also important for the automatic agent is the calendar manager. The agent should only bid for tasks that are compatible with the user's calendar. So the agent needs to check the calendar and when it gets the assignment for a task it inserts the calendar dates to the calendar. To improve the probability of success, we inserted the concept of multi-bids, this is, an automatic agent can bid for different tasks simultaneously. These bids are bundled to a multi-bid. If one of these bids leads to an agreement, the other bids of the multi-bid are invalid. When a bid for an English auction is part of the multi-bid, the other tasks are *blocked* for the time the bid is the actually best bid of the auction. When it receives the task, the other bids are cancelled, otherwise the bid for the English auction will not be the best bid anymore an so it will be removed from the multi-bid.

This possibility for multi-bids depends on the market attitude. In our system, the agent that offers a task can decide whether she allows multi-bids or not. The default adjustment is not to allow multi-bids, because in this case a bid is valid and cannot be made invalid.