Preparing Semantic Agents for an Unsuspecting and Unreliable World

Luke K. McDowell
Computer Science Department
U.S. Naval Academy
572M Holloway Rd
Annapolis, MD 21402
lmcdowel@cs.usna.edu

Introduction

Semantic web-enabled agents offer tremendous promise for enabling automated coordination and negotiation amongst diverse participants. Ideally, a user should be able to quickly instruct his personal agent to achieve some goal, e.g., to schedule a meeting so that at least one representative of each division of the company would attend. This agent would then negotiate with the agents of all the invited participants, receiving prompt and definitive replies, and quickly reach consensus on an acceptable time. The original user would be involved again only when the goal was achieved.

In reality, many problems are likely to arise:

1. **Goal Specification:** The user may know precisely what he wishes to have done, but instructing a general-purpose software agent to achieve this goal can be very complex.
2. **Agent Proliferation:** In a typical organization today, very few participants will know what a software agent is, much less have one that could act on their behalf.
3. **Participant Reliability:** Even if the user's agent is able to directly interact with the participants, many of them are likely to not respond, due to confusion about how to do so or general busyness. Moreover, those that do properly respond may later need to change their response, a situation that the user's agent may not be prepared to handle.
4. **Evolving Goals:** Finally, even the originating user may be uncertain about his precise goals. For instance, he may decide to invite new participants later, or to modify the goals after seeing the initial responses. Such changes pose technical challenges for the agent and may render its earlier actions irrelevant or even counter-productive.

Collectively, these issues pose serious problems to the wide-scale deployment of useful, flexible semantic agents. In previous work, we examined the first issue by considering how ordinary people could specify flexible, explainable semantic agents (McDowell, Etzioni, & Halevy 2004). In this paper, we tackle the other three issues described above.

In particular, we examine these issues in the context of our deployed system for *semantic email agents* (E-Agents) (McDowell *et al.* 2004). E-Agents provide a good testbed for examining these challenges because they offer the potential for managing complex goals and yet are intended to be used by a wide range of untrained people. E-Agents support the common task where an *originator* wants to ask a set of *participants* some questions, collect their responses, and ensure that the results satisfy some set of *constraints*. In the simplest case (which we assume here), the E-Agent will *accept* each response so long as the constraints remain possibly satisfiable; otherwise, the response is *rejected*. We have authored and deployed[1] a number of E-Agents for tasks such as collecting RSVPs, giving tickets away (first-come, first-served), scheduling meetings, and evenly distributing people into K sets (e.g., for committee assignments).

Our contributions are as follows. First, the next section explains how E-Agents solve the agent proliferation problem through the use of dual text/RDF email messages that can be handled either by a human participant or by an agent acting on their behalf. We then introduce a fundamental notion of an *eager response* that represents a new response or a change that a participant is willing to make. In the following sections, we discuss the problems of participant reliability and evolving goals, focusing particularly on how an agent can exploit its knowledge of relevant eager responses to better satisfy the originator's and participants' goals. We explore the computational complexity of these problems and demonstrate how they can be solved in polynomial time in many common cases. In addition, we explore several additional reasoning problems where semantic knowledge can assist the operation of the agent. These results both greatly increase the usefulness of E-Agents as well as identify a number of usability and functionality issues that apply to a much broader range of semantic agents.

Interacting with Agentless Participants

Currently, semantic agents are in use by only a very small number of "early adopters." If an originator's agent could only interact with people who were similarly equipped, such agents would have very limited applicability. Thus, to succeed semantic agents must be usable even when the participants have no experience with or software installed for them.

E-Agents meet this challenge with a combination of techniques. First, the initial communication with a participant is

[1] See http://www.cs.washington.edu/research/semweb/email. These agents may be used without needing to install any software.

via an ordinary email message that contains both a textual portion (for viewing by the human participant) and an RDF portion (for use by the participant's agent, if any). Participants may respond by simply replying to the message and filling out an included text form. Their responses are then converted into RDF at the server with a mapping from each field in the form to an unbound variable in an associated RDQL query. Alternatively, if the participant has an agent, that agent may answer the RDQL query for the participant and respond via the email form.

Instead of email replies, having participants respond via following a link to a web form is another attractive option and is also supported by our E-Agent server. For any semantic agent, the key issue is enabling naive participants to easily respond while readily exploiting agent to agent communication where possible.

Identifying Eager Responses

The remainder of the paper will focus on helping an agent deal with changing responses or goals. In either case, a key part of the agent's ability to react will depend upon its understanding of further changes that participants are willing to consent to:

Definition 0.1 (**eager response**) A response r is *eager* if some participant p is willing to respond with r, if asked. □

Agents can use a variety of techniques to try to estimate the set of eager responses for each participant. For instance, by reasoning about a participant's scheduling information from a calendar server, web page annotations, or other RDF-based sources, the agent may be able to *predict* that certain meeting times are likely (or unlikely) to be possible for that participant. We focus here on the simpler *recognition* task: Suppose a participant makes an initial, first-choice response, but that response is rejected. (Recall than an E-Agent will accept or reject every response it receives.) The participant may or may not respond with a second choice. In either case, it is often reasonable to assume that the participant would be willing to re-submit his original response. For instance, if a participant's request for concert tickets is rejected because not enough tickets remain, the agent could recognize this as an eager response to be revisited if more become available later.

Of course, as time passes not all original choices continue to be valid. There are many interesting possibilities for reasoning about when a rejected response is still "eager" based on later responses, elapsed time, calendar information, inferred preferences, etc. For simplicity, we assume in this paper that any rejected response can be counted on as an eager response. Future work should consider more sophisticated and/or probabilistic methods to predict and recognize eager responses for E-Agents or other semantic agent systems.

Handling Fickle & Nonresponsive Participants

Ideally, participants would respond to an agent's request promptly and definitively. In reality, some participants may respond belatedly or never, or may wish to change their response later. Both problems significantly hamper an agent's ability to optimally pursue its goals.

The first problem can be partially addressed by predicting responses that have not arrived, using the reasoning discussed in the previous section. The second problem, of changed responses, poses different challenges and opportunities. For instance, suppose a participant indicates that he will attend a meeting, but later changes his response. This change may create problems – for instance, causing the meeting quorum constraint to no longer be satisfied. If, however, this meeting were a space-limited seminar, this change could be useful, e.g., to permit a different, previously rejected, participant to attend. Note though that this latter participant will only benefit if the agent can detect that such a change is possible and beneficial.

These challenges and opportunities can both be addressed by exploiting the eager responses of Definition 0.1. More formally, we have the following definition:

Definition 0.2 (**eager satisfiability**) Let Λ be an E-Agent with current state D (representing the responses that have been accepted by the agent) and constraints C_D on D. Let E be a set of eager responses, where each $e \in E$ may either add a new response to D or modify an existing response in D. We say that D is *eager satisfiable* with respect to E and C_D if there exists some set $F \subseteq E$ such that changing D by F makes D satisfiable with respect to C_D. □

Intuitively, an E-Agent Λ is *eager satisfiable* if there is some combination of eager responses that could be solicited in order to make the overall E-Agent's goals satisfiable.

Solving the eager satisfiability problem enables us to address each of our earlier problems. After a response change, the agent can reason about previously rejected responses to determine which such eager responses should now be accepted to make the originator's goals satisfiable or to enable more participants to have their first choice. Unfortunately, solving this problem is intractable in general:

Theorem 0.1 *Let Λ be an* E*-Agent with current state D, eager set E, and constraints C_D that allow conjunction and disjunction of binary predicates. If Λ has N participants, then determining* eager satisfiability *is NP-complete in N.*

This represents a significant problem, since we would like E-Agents to be able to scale up to at least hundreds of participants, e.g., to support company-wide meetings or large program committee organization. Fortunately, in the common case where C_D is *bounded* (which includes all of the examples discussed in this paper), then this problem is solvable in time polynomial in N. Intuitively, the constraints C_D are bounded if what matters is the number of participants that belong to each of a fixed number of groups, rather than the specific participants in each of these groups. For instance, for our example meeting scenario all that matters is the number of people from each company division that can attend a certain meeting time, not the specific responses of each person. The constraints may also make finer distinctions and still be bounded (for instance, to distinguish between managers and regular employees), so long as there is a fixed number of groups of interest.[2]

[2]See (McDowell *et al.* 2004) for a more formal definition and some related applications of bounded constraints.

Supporting Evolving Goals

Finally, an intelligent agent must be able to deal with changes that the originator wishes to make to the agent's goals, including a change to the set of participants. The need for such modifications will inevitably occur due to originator error, changing circumstances, or the emergence of new information. In addition, strong support for such features greatly increases the utility of the agent – if goal changes can be smoothly and competently performed by the agent, the originator can deploy the agent both earlier (before goals are certain) and more often. How, though, can such we support such functionality?

First, the agent must be able to interpret previous responses in light of the new goals. For instance, if a modified meeting goal requires a certain number of `Managers` to attend, the agent may need to translate earlier responses where participants identified themselves as `GroupHeads` or `ProjectLeads` to the new terminology. On a related note, if the originator makes other changes, such as moving the meeting location, the agent needs different kinds of reasoning to determine if previous responses are still valid, or must be re-confirmed. For both kinds of reasoning the agent will need appropriate semantic knowledge and possibly some domain-specific rules. For instance, responses might be deemed still valid if the agent can prove that a new meeting location is within a mile of the previous location.

Second, the agent must be able to reason about the satisfiability of the new goals and the impact of the agent's past actions. Assuming that the goals are initially satisfiable, any such goal change can be classified into exactly one of the following categories:

1. **Optimality-preserving:** After the change, the goals are still satisfiable. Also, the agent's prior actions were optimal relative to the new goals, i.e., it would have chosen the same actions if initially invoked with these new goals.
2. **Satisfiability-preserving:** After the change, the goals are still satisfiable, but some of the agent's prior actions were not optimal relative to the new goals.
3. **Eager-satisfiability-preserving:** After the change, the goals are not immediately satisfiable, but are eager-satisfiable based upon known eager responses. This also implies that the agent's prior actions were not optimal.
4. **Unsatisfiable:** The goals cannot be satisfied given the current responses, even if eager responses are considered.

The ability of an agent to classify a goal change into one of these categories is a significant feature from the perspective of both the originator and the agent. For the originator, being able to immediately see the implications of the goal change may affect his decision as to whether or not to apply the change. Note that the originator's leeway with such goal changes is greatly increased if his agent can reason about the possible impact of incorporating eager responses. For the agent, understanding which category a goal change is in provides immediate direction regarding what actions to take, if any, following such a change.

We now briefly discuss the computational complexity of detecting each of the above categories for E-Agents. Determining whether a goal change yields goals that are still satisfiable (and hence the change was either optimality-preserving or satisfiability-preserving) is straight-forward if the goal constraints are bounded. Distinguishing between these two classes (to evaluate optimality of past actions) is more complex, but can still be done in polynomial time by simulating the past responses. Determining if a goal change is eager-satisfiability-preserving can clearly leverage some of the same reasoning mechanisms that were discussed for dealing with response changes, and such reasoning will be polynomial time under the same conditions as discussed in the previous section.

Related Work and Conclusions

To be effective in the real world, semantic agents must be able to easily and effectively interact with a wide range of participants. In this paper, we have outlined a number of challenges to this vision and explained concrete approaches to overcoming them. In particular, we demonstrated how messages that are actionable by either humans or agents can solve the agent proliferation problem. We also introduced the fundamental notion of *eager responses* and explained how such responses can address the problems of changing responses and goals. We demonstrated that reasoning about such responses is intractable in the worst case, but can often be done in polynomial time. Finally, we outlined the need for additional semantic reasoning procedures to predict and interpret participant responses.

While we focused on the context of E-Agents, our results are relevant to many other agent systems. For instance, the appointment scheduler of Berners-Lee, Hendler, & Lassila (2001) or the meeting scheduler of Payne, Singh, & Sycara (2002) both may have the need to deal with agentless participants that may not respond immediately or definitively. Likewise, while the travel planning agent of McIlraith, Son, & Zeng (2001) is designed only to interact with suitably designed web services, applying the ideas of this paper would enable such a system to flexibly work with many additional participants (such as human travel agents). Finally, preparing for possible goal changes by proactively collecting additional information (such as potential eager responses) would greatly increase the utility of many agent systems.

Acknowledgements

This work was partially supported by the Naval Academy Research Council and ONR grant N0001405WR20153.

References

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web. *Scientific American*.

McDowell, L.; Etzioni, O.; Halevey, A.; and Levy, H. 2004. Semantic email. In *Proc. of the Thirteenth Int. WWW Conference*.

McDowell, L.; Etzioni, O.; and Halevy, A. 2004. The specification of agent behavior by ordinary people: A case study. In *Third International Semantic Web Conference*.

McIlraith, S. A.; Son, T. C.; and Zeng, H. 2001. Mobilizing the semantic web with daml-enabled web services. In *Proceedings of the 2001 Semantic Web Workshop*.

Payne, T.; Singh, R.; and Sycara, K. 2002. Calendar agents on the semantic web. *IEEE Intelligent Sys* 17(3):84–86.