Hermes: A Methodology for Goal Oriented Agent Interactions

Christopher Cheong
chris@cs.rmit.edu.au

Michael Winikoff
winikoff@cs.rmit.edu.au

RMIT University
Melbourne, Australia

ABSTRACT

We present the *Hermes* approach for goal-oriented interaction which includes a methodology for designing goal-based interactions and a process for mapping design artefacts to an executable implementation.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques

General Terms

Design

Keywords

Interaction Protocol, Goal-Oriented, Agent-Oriented Software Engineering

1. INTRODUCTION

Currently, agent interactions protocols are expressed in notations such as Agent-UML [4], Petri nets [6], or nite state machines. However, these approaches are not well-suited to agents that are autonomous and proactive. Interaction protocols are at a low level of abstraction and are message-centric in nature since they are de ned in terms of legal message sequences.

This results in a number of drawbacks for the agent paradigm. The primary disadvantages are that the protocols are mechanistic and restrict the autonomy of intelligent agents. Since agents are autonomous and able to independently pursue goals and recover from failures, their interactions should exploit, rather than limit, these characteristics. Further disadvantages are that the exibility and robustness of the interactions are limited (as the degree of exibility and robustness depend on the number of legal message sequences); where exibility refers to multiple ways to successfully achieve an interaction and the ability to take *shortcuts* (i.e. by-passing already completed parts of the interaction), and robustness is the ability to recover from and persevere through failures in the interaction.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

We propose the concept of *goal-oriented interaction* which is better suited to the agent paradigm's goal-oriented nature. Goal-oriented interactions are comprised of *interaction goals* (IG) and temporal constraints. The interacting agents determine how interaction goals are achieved and are restricted by the temporal constraints placed on the IGs. Interactions between agents occur because the agents involved have certain goals to achieve, and the interactions are a means of achieving the agents' goals. In this manner, the legal message sequences *emerge* from the interaction, as opposed to interaction protocols in which legal message sequences must be explicitly de ned. Emergent message sequences allow for a greater number of message sequences than what could be explicitly stated which results in greater exibility and robustness in interactions.

We aim to devise a *practical* approach that can be used to develop exible and robust interactions in agent systems which specifically includes a design methodology and execution mechanisms. We thus introduce the *Hermes*[1] methodology. Hermes is a domain independent methodology which provides a systematic approach for creating goal-oriented interactions and thus moves away from message-centric protocols. Hermes includes both a methodology for designing agent interactions in terms of interaction goals (described very brie y in the next section, and see [1]), and a mapping from the design artefacts produced to an implementation in a plan-goal based agent programming language such as Jadex[2] [2].

2. GOAL ORIENTED INTERACTION DE SIGN

An overview of the Hermes design process is shown in Figure 1. The rst two steps in the methodology are centered around *interaction goals* (or *IGs*), which are high level goals that must be achieved for the interaction to be successful. The rst step is to identify IGs whilst the second step involves re ning[3] and organising the IGs into a hierarchy. At its apex, the hierarchy has a single IG which captures the overall goal of the interaction as a whole.

As an example, we use an e-commerce protocol based on the NetBill [7] protocol in which a Customer purchases goods online from a Merchant. Figure 2 shows a hierarchy for such a protocol where the circles represent IGs. The *Trade* IG at the apex captures the overall intent of the interaction and is decomposed into smaller sub-IGs, such as *Agree* and *Exchange*.

[1] In Greek mythology, Hermes was an Olympian god who acted as the herald of the gods and served as their messenger (http://www.pantheon.org).
[2] http://vsis-www.informatik.uni-hamburg.de/projects/jadex/
[3] This includes decomposing identi ed IGs into smaller sub-IGs

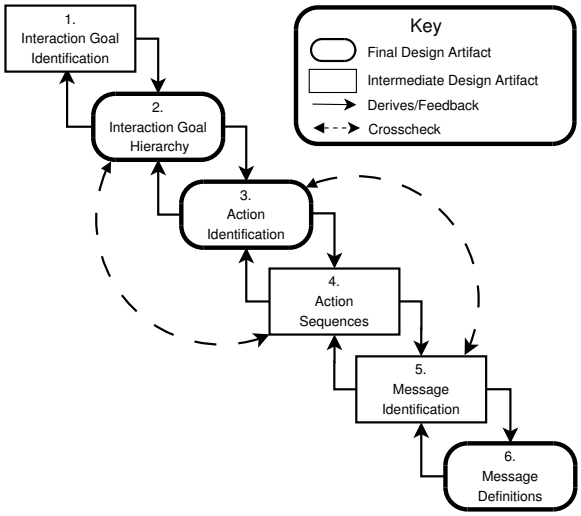

Figure 1: Hermes Methodology Overview Diagram

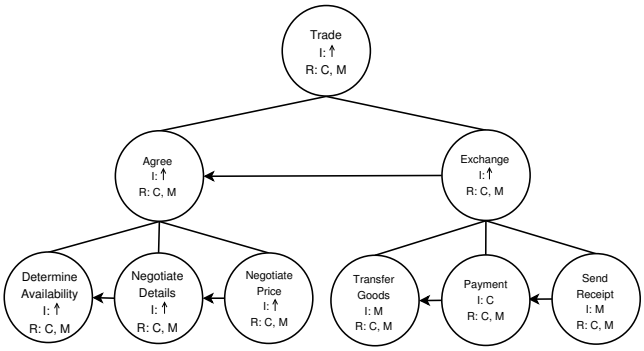

Figure 2: Interaction Goal (IG) Hierarchy Diagram

In Figure 2, the roles involved are shown in the circles as *R: C, M*, denoting that a particular interaction goal involves the *Customer* and *Merchant* roles. An *initiator*[4] role is also shown. Valid initiators are specified as one of the roles involved in a particular IG (e.g. *C* or *M*) or as ↑ if it is an *inherited* role, i.e. the parent interaction goal's initiator.

The interaction goal hierarchy is effectively a goal-tree, similar to those used in agent-oriented methodologies such as MaSE [3] or Prometheus [5].

The middle two steps of the methodology deals with actions. An *action* is a discrete step towards achieving an interaction goal which is taken by a single agent. For each leaf-node IG, actions which can be used to achieve the IG are identified. The actions are assigned to roles and are structured in a flexible sequence of execution. For example, in the *Negotiate Price* IG, the Customer starts by using a *Propose Price* action, after which the Merchant will use a *Consider Price* action, and so on.

Once action sequences have been determined, the two remaining steps of the methodology deals with identifying inter-agent messages and their format. Although Hermes provides guidelines to assist with identifying the messages, details of the message format are typically specific to the application and the implementation platform, and thus Hermes does not provide any guidelines for developing the message format, nor any constraints on the message format: one could choose to use KQML, FIPA, SOAP, or the message types provided by the implementation platform (for a closed agent system).

[4]The role which initiates and is initially responsible for a particular IG of the interaction

3. DISCUSSION

The discussion above did not cover the failure handling aspects of the methodology. There are two types of failure in Hermes [1]: an action can fail, in which case it is usually appropriate to try other actions in order to achieve the current interaction goal (*action retry*); or, an interaction goal can fail, in which case either the whole interaction fails or we can try and re-solve a previous IG in a different way (*rollback*). For example, if the merchant and customer cannot agree on a price, then the details of the produce could be re-negotiated before re-attempting to agree on a price.

Agent interactions are only one part of creating an agent system. As such, we intend to integrate Hermes with an agent methodology, such as Prometheus [5]. The design methodology and notation will also require further refinement as we undertake research into adapting Hermes to function with protocols which involve many agents, including non-goal-based agents. Additionally, since we aim for Hermes to be practical, tool support is an important area for future work. Other, longer term, areas for future work include looking at the verification of goal-oriented interactions, and an experimental evaluation of the approach.

Acknowledgements

We would like to acknowledge the support of Agent Oriented Software Pty. Ltd. and of the Australian Research Council (ARC) under grant LP0453486.

4. REFERENCES

[1] C. Cheong and M. Winikoff. Hermes: Designing goal-oriented agent interactions. In *Proceedings of the Sixth international Workshop on Agent-Oriented Software Engineering (AOSE)*, July 2005.

[2] C. Cheong and M. Winikoff. Hermes: Implementing goal-oriented agent interactions. In *Proceedings of the Third international Workshop on Programming Multi-Agent Systems (ProMAS)*, July 2005.

[3] S. A. DeLoach, M. F. Wood, and C. H. Sparkman. Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):231–258, 2001.

[4] M.-P. Huget and J. Odell. Representing agent interaction protocols with agent UML. In *Proceedings of the Fifth International Workshop on Agent Oriented Software Engineering (AOSE)*, July 2004.

[5] L. Padgham and M. Winikoff. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley and Sons, 2004. ISBN 0-470-86120-7.

[6] W. Reisig. *Petri Nets: An Introduction*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1985. ISBN 0-387-13723-8.

[7] M. Sirbu and J. D. Tygar. NetBill: An Internet Commerce System Optimized for Network-Delivered Services. *IEEE Personal Communications*, 2(4):34–39, August 1995.