Using Conceptual Graphs to Represent Agent Semantic Constituents

Lois W. Harper, Harry S. Delugach

Department of Computer Science
N300 Technology Hall
{lharper, delugach} @ cs.uah.edu
University of Alabama in Huntsville
Huntsville, AL 35899 USA

Abstract. This paper develops two agent knowledge bases in conceptual graph form, one using the KD45 underlying logical model for *belief* and one without any underlying logical model for *belief*. Action-attitudes in the knowledge bases provide contexts that represent the agents' mental attitude towards, and willingness to act upon information in the knowledge bases. Preconditions for communication acts are also represented in the knowledge bases as well as mental attitude changes following communications. Conceptual graphs are a flexible and extendable form of knowledge representation that is used to capture and represent semantic constituents of communications in a form that may be used by software agents. The knowledge base representations in this paper provide software agents a perspective from which they may reason about the communicating agent's *beliefs* and communication actions.

1 Introduction

Software agent technologies are being directed towards enabling heterogeneous computing platforms in open environments to communicate, share resources and cooperatively delegate tasks. Software agents provide developers a high-level abstracted view from which to more easily solve problems in these programming environments. [1] *Intelligent* software agents are often characterized as holding mental attitudes, such as *belief*, towards their knowledge. These agents may use an agent communication language (ACL), such as KQML or the FIPA ACL, to convey their mental attitudes towards communicated knowledge. [2, 3]

An underlying assumption with ACLs is that the communicating software agents 'speak' the same ACL. [4] This presents difficulties for software agents designed to converse in different ACLs, or to converse with different dialects of the same ACL. Other approaches have been taken towards establishing common vocabularies between software agents. Some examples are:

- [5] The World Wide Web Consortium (W3C) Ontology Web Language (OWL) with the Resource Description Framework (RDF) allows web-based ontologies to be established as meta-languages, providing web agents shared vocabularies. The DAML-S ontology supports locating web services. [6]

- Gmytrasiewicz et al.[7] proposes that software agents negotiate and establish a common vocabulary at runtime, while Reed et al. [8] propose that communicating agents share a common set of semantic primitives but negotiate the final meaning of these primitives at runtime.
- Sycara et al. [9] use matchmaker agents to support advertising, requesting and matching agent services using a communication language in addition to their ACL, called the Agent Capability Description Language (ACDL) LARKS. Matchmaker agents are not information brokers, so after LARKS is used to locate services, the matched agents must still intercommunicate to use services.

These approaches address agent vocabularies in different semantic constituents: ACLs for agent communication protocols; negotiated semantics for heterogeneous agent messages; matchmaker agents for locating agent services, and web ontologies for establishing and using web-based domain knowledge.

Once a common vocabulary of any type has been established, the expected use of this vocabulary in dynamic situations, such as communication protocols, needs to be represented to software agents. These representations may assist heterogeneous agents in reasoning about and overcoming other communication barriers. Hodgson lists these requirements for representation schemes in AI, emphasizing that the last point is most important [10] :

- The representation scheme should "permit sufficient internal organization so that the type of object (problem) being represented can be easily classified."
- The representation scheme should "be flexible enough so that decompositions of problems can be expressed naturally in terms of the representation."
- The representation scheme should "use as few special structures as possible."

For example, when representing communication protocols in the FIPA ACL, the type of problem is a procedure, and the problem is expressed in terms of the agents' mental attitudes towards knowledge, using the Kripke's KD45 possible world semantics for the mental attitude *belief* [3]. Delugach has developed an extension to conceptual graphs to capture dynamic behaviors [11]. Sowa has discussed the use of context in conceptual graphs for modal logic representation [12, 13]. This suggests that conceptual graphs are a notational framework that will satisfy the representation requirements listed above when representing FIPA agent communications.

2 A Kripke Model of Belief in Conceptual Graph Form

Many theories of rational agency have been developed to allow developers to reason about the behaviors of *intelligent* (*rational*) agents. Four well known theories of rational agency, as described in [14, 15] are:

- Cohen and Levesque [16] – temporal logic + belief (KD45) + goal (KD)
- Moore [17] – dynamic logic + knowledge (S5)
- BDI [18] – temporal logic + belief (KD45) + desire (KD) + intention (KD)
- KARO [19] – dynamic logic + belief (KD45) + wishes (KD)

These theories use the KD45 modal logic model for *belief*. The FIPA ACL's logical model for *belief* is also the Kripke KD45 logical model.[3] Software agents that reason about communicating in the FIPA ACL may therefore benefit from access to a KD45 model of *belief*.

2.1 KD45 Modal Belief Logic

The following discussion is taken from [20, 21]. Modal *belief* logics typically add a modal operator for *belief*, often denoted as **B**, to first-order logic. The rule "If α is a well formed formula, then $\mathbf{B}\ \alpha$ is also a well formed formula" is also added.

A Kripke modal frame is a structure **M** = < **W**, **D**, **R**, **F**> where: **W** is a non-empty set of possible worlds; **D** is a non-empty domain of individuals; **R** is a binary accessibility relation on **W**; and **F** is a state dependent interpretation function. A formula α is satisfiable with respect to the modal structure if there exists a triple (M, w, f) such that (M, w, f) |= α. A formula α is valid if for every triple (M, w, f) it is the case that (M, w, f) |= α. For example, ($\mathbf{B}\alpha \vee \neg \mathbf{B}\alpha$) is valid, and ($\mathbf{B}\alpha \vee \mathbf{B}\neg\alpha$) is satisfiable but not valid. [20]

The term 'KD45' identifies four formulas for *belief*, shown in the following table. Kripke's insight was to show that these accessibility relations correspond to these formulas.

Table 1. KD45 Formulas and Corresponding Frame Relationships

Axiom	Formula	Description	Accessibility Relation
K	$(\mathbf{B}\ \alpha \wedge \mathbf{B}(\alpha \rightarrow \beta)) \rightarrow \mathbf{B}\ \beta$	Beliefs are closed under logical consequence.	
D	$\neg\mathbf{B}$***false***	Falsehoods are not believed.	serial
4	$\mathbf{B}\ \alpha \rightarrow \mathbf{B}\ \mathbf{B}\ \alpha$	Beliefs are closed under positive introspection.	transitive
5	$\neg\mathbf{B}\ \alpha \rightarrow \mathbf{B}\ \neg\mathbf{B}\ \alpha$	Beliefs are closed under negative introspection.	Euclidean

2.2 Mental Attitudes and Action Attitudes

The FIPA specification states that software agents using the FIPA-ACL will possess the three primitive mental attitudes of *belief*, *uncertainty* and *choice*, formalized in the FIPA Semantic Language by the modal operators **B**, **U** and **C** respectively. An FIPA agent is *uncertain* (**U**) about a proposition *p* if it considers that *p* is more likely to be true than not true. The FIPA mental attitude *choice* (**C**) represents a goal state concerning a proposition.[3]

An autonomous agent may *believe* a proposition α, $\mathbf{B}(\alpha)$, but may choose not to take action on that *belief*. For example, the agent may receive a query concerning the truth-value of a proposition that it *believes* is true, but may choose not to answer the query. As intelligent and autonomous software processes, software agents will make decisions based on their input, existing knowledge, interaction protocols and other established plans. [22-24] These factors are not directly addressed by an agent communication language, but nonetheless will contribute towards determining when and what communication acts may take place. The predicate variable **A**, for *action*, is used in this paper to represent these factors that, together with the mental attitude of *belief* differentiate four mutually exclusive "action-attitudes". These are shown in Table 2.

Table 2. Four Mutually Exclusive Action-Attitudes

	Believes; B(α)	**Does Not Believe; ¬B(α)**
Will Act; A(α)	Action-Attitude 1 (AA1)	Action-Attitude 2 (AA2)
Will Not Act; ¬A(α)	Action-Attitude 3 (AA3)	Action-Attitude 4 (AA4)

2.3 A Conceptual Graph Model of the Kripke KD45 Frame

In this section we develop a representation of the KD45 model of *belief* in conceptual graph form. The FIPA Communicative Act Library Specification does not specify any agent implementation. [3] Our model is simply developed for this example of showing how a model of *belief*, in a form accessible to software agents, may assist agents in reasoning about communication acts.

- Since the FIPA ACL uses the Kripke KD45 model of *belief*, we see that these action attitudes may correspond to possible worlds in a Kripke *belief* structure.
- Since communication acts may be initiated by agents that have no knowledge of the truth-value of a proposition, such as in the FIPA query-if performative, we let the mental attitude *choice*, **C,** be a subworld of AA2. [3]
- Because the preconditions for communication in the FIPA ACL ensure that no proposition is conveyed by an agent that is uncertain about the truth-value of the proposition, we let the mental attitude *uncertain*, **U,** be a subworld of AA3. [3]
- Because a speech act is an action according to the speech act theories of [25, 26], preconditioned by *beliefs* specified in [3], we do not use AA4 in our model.

A Kripke frame with three worlds, each an action-attitude described above, and with the KD45 accessibility relation is then:

- A set W whose elements are all possible worlds (AA1, AA2, AA3)
- The accessibility relations: R(AA1, AA2), R(AA2, AA1), R(AA2, AA3), R(AA3, AA2), R(AA3, AA1), R(AA1, AA3)

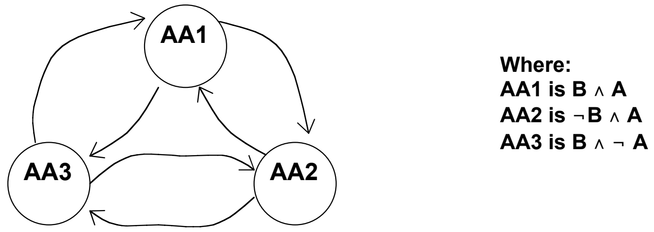

Fig.1. KD45 Frame

The KD45 Kripke frame in Figure 1 is represented in conceptual graph notation in Figure 2. Delugach has extended conceptual graph notation to represent dynamic behaviors, such as the assertion of temporal knowledge, using a relationship called a demon [11]. When activated, the demon will assert each of its output concepts and retract each of its input concepts. The demons in the conceptual graph below are activated by changes in action-attitudes held by the agent, with respect to a proposition. Within the context of possible world semantics, these demons allow an agent to change state between possible worlds. The previous world (action-attitude) is retracted with respect to a proposition, and a new world (action-attitude) is asserted.

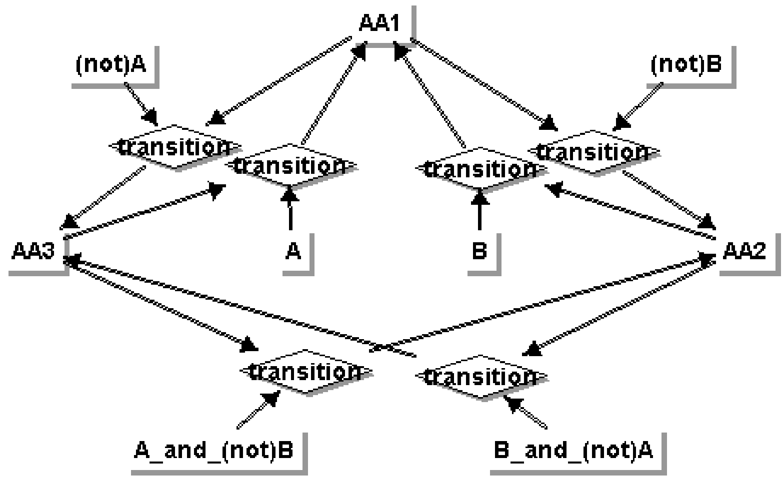

Fig. 2. CG Representation of KD45 Frame with Action-Attitudes as Possible Worlds

The possible worlds AA1, AA2 and AA3 in Figure 2 are represented by concepts that are action-attitudes within which propositions are nested. When an FIPA agent *believes* a proposition and is willing to act on that proposition, then the agent is residing in possible world AA1 with respect to that proposition. Similarly, if the agent has no knowledge of the truth value of a proposition but is willing to act; the agent resides in possible world AA2. If the agent has a *belief* (uncertain or certain) with respect to a proposition but is unwilling to act, the agent resides in possible world AA3.

3 A Rules-Based Model of Belief in Conceptual Graph Form

As with the FIPA ACL, the KQML ACL is a communication protocol expressed in terms of an agent's mental attitudes towards theirs and other agents' knowledge. However, the KQML ACL uses the mental attitudes *belief*, *knowledge, desire* and *intention*. The KQML mental attitude *knowledge* functions similarly to the FIPA mental attitude *belief*. No semantic models for mental attitudes are specified in KQML, but the language used to describe speech acts restricts the way mental attitudes can be used in speech acts. [27]

Initially, the KQML had no formal specification, although this was subsequently addressed. [27] As a result, there are different implementations of KQML that cannot be assumed compatible with each other. Every KQML implementation will necessarily meet some constraints on mental attitudes as an artifact of software design and implementation, although these constraints may not be explicitly stated.

The absence of an underlying logical model for mental attitudes in KQML is not a problem for our approach. We start with the same four action attitudes listed above in Table 2. A rule set can be established to govern the mental attitude *knowledge* for the particular version of KQML ACL being represented. We represent the resulting rule sets as IF-THEN statements in conceptual graph form, as shown in Figure 3. (There are several possible conceptual graph representations for IF-THEN statements, different from that shown here.)

The action-attitudes in Figure 3 will also form a context for each proposition in the KQML agent's knowledge base. We do not assume that action-attitude AA4 is not required. The particular implementation of KQML will need to be examined first, then the rule sets structured to reflect the particular KQML implementation being represented.

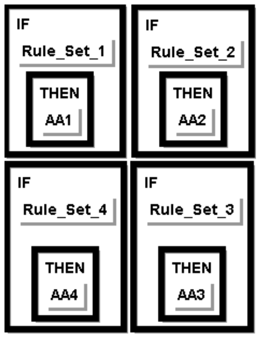

Fig. 3. Rule Sets to Define Action Attitudes

4 Knowledge Bases with Action Attitudes

The preconditions that determine when a FIPA agent may issue a communication act are stated in terms of that agent's mental attitudes towards a proposition. The proposition is located in the content field of the ACL message. [3] The knowledge base of a FIPA agent will need to correlate the conveyed propositions to the mental attitudes that the agent takes towards the propositions. The knowledge base may also correlate the mental attitudes of other communicating agents take towards these same propositions, when that information is available.

In conceptual graph form, this can be accomplished by placing propositions within the context of the agent's action-attitudes. In Figure 4, 'Allen' and 'Burt' identify two FIPA agents. A co-referent link identifies a proposition **t** that is in Allen's knowledge base and believed by Allen to also be in Burt's knowledge base. Figure 4 shows that Allen believes that he and Burt have the same action-attitude (AA2) towards proposition **t**.

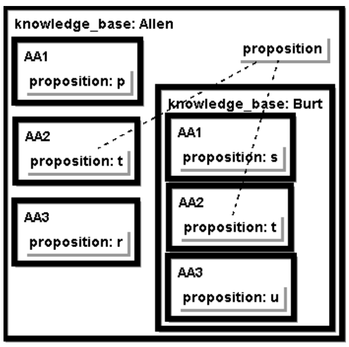

Fig. 4. Knowledge Base of Agent with Action Attitudes Providing Context

Allen's knowledge base in Figure 4 shows not only his attitudes towards propositions in his knowledge base, but also what Allen thinks are Burt's attitudes towards propositions in Burt's knowledge base. As a result, the knowledge base describes preconditions placed on FIPA communication acts, expressed in terms of mental attitudes taken by software agents towards propositions. For example, Allen may issue the FIPA ACL *inform* communication act to Burt with respect to a proposition *p*, only when Allen has action-attitude AA1 with respect to proposition *p*, and Allen does not *believe* Burt holds action-attitudes AA1 or AA3 with respect to proposition *p* (i.e., Burt has no knowledge of the truth-value of proposition *p*).

The rational effect of FIPA ACL communication acts is also expressed in terms of the agents' mental attitude towards propositions. If Allen issues the FIPA *confirm* communication act to Burt with respect to proposition *p*, a rational effect is that proposition *p* is asserted in Burt's knowledge base. Allen's model of Burt's knowledge base is not required to be accurate however; two agents will not have complete and accurate representations of each other's knowledge bases.

Although the KD45 logical model is specified for the FIPA ACL and no underlying logical model is specified for the KQML ACL, the knowledge bases of both types of agents may be represented using action-attitude contexts. The action-attitudes are established as discussed in the previous section. A step towards establishing common semantic knowledge between the two different types of agents is to determine whether the *knowledge* mental attitude of the KQML agent plays the part of the *belief* mental attitude of the FIPA agent. If these are accepted as being close, as they are in [8], then the action-attitudes of the FIPA agent may be considered to be very close to those of the KQML agent.

Since FIPA agent Allen contains within his knowledge base a representation of KQML agent Burt's knowledge base, Allen may use his representation of Burt's knowledge base to reason about communication with Burt. Allen may also evaluate whether he *believes* that common knowledge exists between Burt and himself. As an example:

If Burt has action-attitude AA2 with respect to proposition *p*, then Burt has an uncertain *belief* of proposition *p*.
If Burt has action-attitude AA2 with respect to proposition *p*, then Burt may not inform another agent of the truth-value of *p*.
Burt has action-attitude AA2 with respect to proposition **t**.
Therefore, Burt may not inform Allen of the truth-value of **t**.

5 Two Examples of Action Attitudes in Agent Knowledge Bases

The use of action-attitude contexts to represent agent attitudes towards propositions is useful both when modeling communication between software agents that use different ACLs, and also with communication between software agents that use the same ACL. In the examples below, 'Allen', 'Burt' and 'Charlie' identify three software agents that use the FIPA ACL. The conveyed information is the sentence "Car is repaired".

5.1 Communication between two FIPA Agents

The sequence of messages between Allen and Burt in this first example is shown in Table 3, where the first communication act is Message 1. This message is the FIPA *query-if* communications act. Its semantic effect is to indicate that Allen is asking Burt if the statement "Car is repaired" is true. The second message is the FIPA *inform-if* communications act. Its semantic effect is to indicate that Burt is telling Allen that Burt *believes* that the statement "Car is repaired" is true.

Although the conveyed propositions, "Car is repaired", are identical, the two FIPA communication acts (*query-if* and *inform-if*) do not convey the same mental attitudes towards that proposition. Burt has no knowledge of the truth-value of the conveyed message; while Allen must *believe* that the statement "Car is repaired" is true. The two communication acts also have different rational effects. Once Message 2 has been issued, Allen may also 'think' that Burt *believes* that the statement "Car is repaired" is true, although Allen may not *believe* "Car is repaired" is true. In Figure 5, depicting Allen's knowledge base following Message 2, Allen is shown as *believing* the statement and also thinking that Burt *believes* the statement.

Table 3. Two FIPA Communication acts

Message 1 – Burt queries Allen	**Message 2 – Allen informs Burt**
(query-if :sender Allen :receiver Burt :content "Car is repaired" :protocol FIPA Query ...)	(inform-if :sender Burt :receiver Allen :content "Car is repaired" :protocol FIPA Query ...)

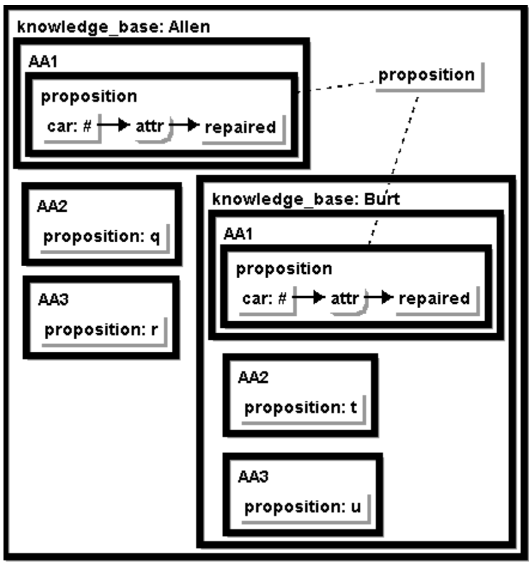

Fig. 5. Knowledge Base of Allen Following Step Two in Table Three

5.2 The FIPA Proxy Communication Act and Agent Cooperation

The FIPA describes the *proxy* communication act as having two strengths, referred to as *strong-proxy* and *weak-proxy*. Both *strong-proxy* and *weak-proxy* employ a third, middle agent to relay information between two agents. The third, middle agent is the proxy agent. *Strong-proxy* occurs when the proxy agent *believes* that the proposition it relays in the *proxy* communication act is true. *Weak-proxy* occurs when the proxy agent does not *believe* that the proposition is relays in the *proxy* communication act is true. [3] The two types of proxy communication acts are illustrated by the communication sequence shown in Table 4. The proxy agent is Burt, who is relaying information between agents Allen and Charlie.

Table 4. Weak and Strong Proxy Communications

	Conversation 1: Strong Proxy	Conversation 2: Weak Proxy
1	Allen calls the repair shop to ask if his car is repaired.	(Same as Conversation 1)
2	Burt answers the phone and hears Allen's question.	(Same as Conversation 1)
3	Burt tells Charlie that Allen is asking if his car is repaired.	(Same as Conversation 1)
4	Charlie asks Burt to tell Allen that Allen's car is repaired.	(Same as Conversation 1)
5	Burt tells Allen that the car is repaired.	Burt tells Allen that Charlie says that the car is repaired.

The two conversations are identical until Step 5. This is because the decision to return a *strong* or *weak-proxy* is determined at run time. Burt is an autonomous agent who may not *believe* that Charlie is to be trusted. As a result, Burt may not *believe* Charlie's answer.

A first problem presented by Burt not *believing* Charlie is that it is a precondition on the FIPA *inform* communication act that an agent issuing an *inform* message *believes* that the information being conveyed is true. The FIPA specification handles this problem by allowing the Burt to reply "Charlie says that the car is repaired." This is a true statement. The *inform* communication act precondition is satisfied and the *proxy* communication act may be completed as shown in Conversation 2.

A second problem is to evaluate whether Allen may detect that a *weak-proxy* has occurred. Specifically, how can Allen determine whether Burt *believes* Charlie's answer? The FIPA Communicative Act Library Specification states that the feasibility preconditions placed on communication acts have two parts; ability preconditions and context-relevance preconditions. Ability preconditions refer to the software agent's ability to execute a communication act. Context-relevance preconditions refer to the relevance of executing a communication act. The context-relevance preconditions correspond to Grice's Quantity and Relation Principles. [3] These principles, shown in Table 5, are two of four cooperation principles identified by Grice. The principles are applied to ACLs because ACL performatives are rooted in speech act theories modeled after spoken human communication. [25, 26]

Table 5. Two Gricean Cooperation Principles, referred to by the FIPA

Grice Cooperation Principle	Purpose
Quantity	Be only as informative as required for the purposes at hand.
Relation	Make only relevant statements.

Software agents that can detect when a Gricean cooperation principle has failed may conclude that a communicating agent is not cooperating. In this simple example, Allen may determine that the Quantity Principle has failed when Burt responds, "Charlie says that the car is repaired", instead of "Car is repaired." Although each statement may be true, more information is returned in the first statement than in the second. The *weak-proxy* may be detected by determining that more information is being returned than is required.

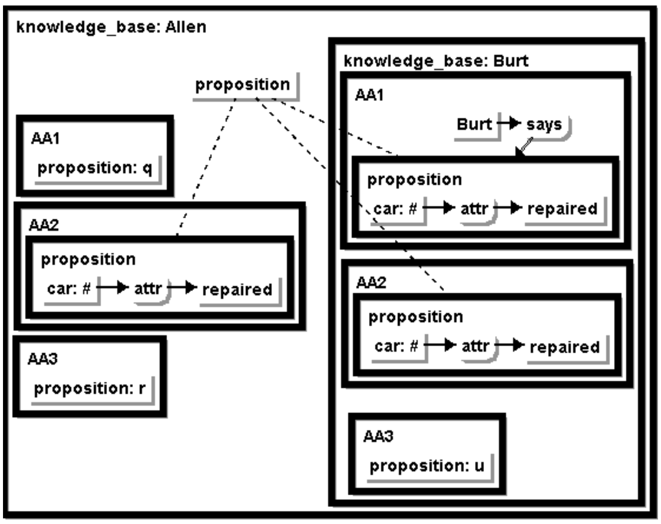

Fig. 6. Allen's Knowledge Base, Following Step Five of Conversation Two

Figure 6 shows that following Step 5 of Conversation 2: (1) Allen is *uncertain* of the truth-value of the statement "Car is repaired", located in AA2; (2) Allen knows that Burt has told Allen, "Charlie says that the car is repaired." By the Gricean Quantity Principle, the additional context for the conveyed statement may indicate that either Burt or Charlie is not cooperating. As a result, (3) Allen may choose to *believe* that Burt also holds AA2 with respect to the statement "Car is repaired."

6 Discussion

Many different formal representations have been used to model software agent constituents, at all stages of their development and implementation. For example, predicate logic has been used to represent facts about 'things' in the agent world, Petri nets or AUML have been used to represent processes and interaction protocols, OWL has been developed to represent object types and ontologies of languages [28], and programming languages such as JAVA have been used to support agent execution across a variety of computing platforms. No one representation form is sufficient for all purposes, and no unifying semantic framework for software agent technologies has been established. [29]

The efforts to establish common semantic frameworks among different types of software agents described at the beginning of this paper indicate that this is a significant goal. If software agents are to reason about themselves, other software agents and their operating environments, these agents will need access to the same types of information that designers use to reasoning about agents and their operating environments.

Although no single formalism is sufficient for every purpose, a form that may integrate several kinds of knowledge may better support agents in reasoning about their and other agent behaviors. This paper explores how representations in conceptual graph form may capture the semantics of modal *belief* logics and dynamic communication processes for software agents.

7 Conclusion

We have shown that communication between intelligent software agents is a complex process that involves semantic constituents in addition to shared vocabularies, as well as a common syntax. In our examples, we have shown the agents communicating to express both factual knowledge and mental attitudes expressed in terms of modal logics. Their knowledge bases developed in our examples show that these types of information may be represented in conceptual graph form, utilizing the demon relation to represent controlled transitions between states in a Kripke KD45 possible world *belief* model, and using concepts as contexts to represent attitudes towards propositions. In doing so, this paper addresses the larger problem identified by [23] and other researchers; which suggests that for intelligent software agents to reason about their own and other agent behaviors, the various types of knowledge used by software developers in designing and implementing software agents may also need to be provided to software agents themselves.

Acknowledgements

The authors would like to thank Lisa Cox and the anonymous reviewers for their comments and suggestions concerning this paper. Their evaluations have substantially improved the clarity of this presentation.

References

1. Wooldridge, M. and N.R. Jennings, *Intelligent Agents: Theory and Practice.* Knowledge Engineering Review, 1995. **10**(2): p. 115-152.
2. Finin, T., et al., *Specification of the KQML Agent-Communication Language*. 1993.
3. *FIPA Communicative Act Library Specification.* http://www.fipa.org/repository/fipa2000.html, 2000.
4. Genesereth, M. and S.P. Ketchpel, *Software Agents.* Communications of the ACM, 1994. **37**(7).
5. W3C, *World Wide Web Consortium.* http://www.w3.org/, 2004.
6. Luck, M., R. Ashri, and M. D'Inverno, *Agent-Based Software Development*. 2004, Norwood, MA.: Artech House, Inc. 208.
7. Gmytrasiewicz, P., M. Summers, and D. Gopal. *Toward Automated Evolution of Agent Communication Languages*. in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. 2002.
8. Reed, C.A., T.J. Norman, and N.R. Jennings, *Negotiating the Semantics of Agent Communication Languages*, in *Computatonal Intelligence*. 2002. p. 229-252.
9. Sycara, K., et al., *LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace*, in *Autonomous Agents and Multi-Agent Systems*. 2002. p. 173-203.
10. Hodgson, J.P.E., *Knowledge Representation and Language in AI*. 1991, Chichester, West Sussex, England: Simon & Simon International Group. 230.
11. Delugach, H.S., *Specifying Multiple-Viewed Software Requirements With Conceptual Graphs.* Journal of Systems and Software, 1992. **19**: p. 207-224.
12. Sowa, J.F., *Conceptual Structures: Information Processing in Mind and Machine*. 1984, Reading, Mass.: Addison-Wesley. 481.
13. Sowa, J.F., *Laws, Facts, and Contexts: Foundations for Multimodal Reasoning*, http://www.jfsowa.com/pubs/laws.htm.
14. Dixon, C., M. Fisher, and A. Bolotov, *Resolution in a Logic of Rational Agency*. Proceedings of the 14th European Conference on Artificial Intelligence. 2000, Berlin.
15. van der Hoek, W. and M. Wooldridge, *Towards a Logic of Rational Agency.* Logic Journal of the IGPL, 2003. **11**(2): p. 135-159.
16. Cohen, P.R. and H.J. Levesque, *Intention is Choice with Commitment.* Artificial Intelligence, 1990. **42**: p. 213-261.

17. Moore, R.C., *A Formal Theory of Knowledge and Action*, in *Readings in Planning*, J.F. Allen, J. Hendler, and A. Tate, Editors. 1990, Morgan-Kaufmann: San Mateo, CA, USA. p. 473-484.
18. Rao, A.S. and M.P. Georgeff, *Modeling Rational Agents Within a BDI-Architecture*, in *Proceedings of Knowledge Representation and Reasoning*, R. Fikes and E. Sandewall, Editors. 1991, Morgan-Kaufmann. p. 473-484.
19. van Linder, B., W. van der Hoek, and J.C. Meyer, *Formalising Motivational Attitudes of Agents: On Preferences, Goals and Committments*, in *Intelligent Agents II*, M. Wooldridge and J. Muller, Editors. 1996, Springer-Verlag.
20. Bacchus, F., *Probabilistic Belief Logics*, in *European Conference on Artificial Intelligence*. 1990. p. 59-64.
21. Huth, M.R.A. and M.D. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*. 2001, New York: Cambridge University Press. 387.
22. Wooldridge, M., *An Introduction to MultiAgent Systems*. 2002, West Sussex: John Wiley & Sons, Ltd. 348.
23. Shen, W., D.H. Norrie, and J.-P.A. Barthes, *Multi-agent Systems for Concurrent Intelligent Design and Manufacturing*. 2001, New York: Taylor Francis Inc. 381.
24. Franklin, S. and A. Graesser, *Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents*, in *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. 1996, Springer-Verlag: Springer-Verlag. p. 21-35.
25. Searle, J.R., *Speech acts: An Essay in the Philosophy of language*. 1969, Cambridge University Press: Cambridge.
26. Austin, J.L., *How to do things with words*. 1962, Oxford University Press: London.
27. Labrou, Y., *Standardizing Agent Communications*, in *Lecture Notes in Computer Science*. 2001. p. 74.
28. Willmott, S., I. Constantinescu, and M. Calisti. *Multilingual Agents: Ontologies, Languages and Abstractions*. in *proceedings First International Workshop on Ontologies in Agent Systems*. 2001. Montreal Canada.
29. Willmott, S., J. Dale, and P. Charlton, *Agent Communication Semantics for Open Environments*. 2002, Agentcities.RTD Project;http://www.agentcities.org/EURTD/Pubs/eurtd.02.willmott.semantics.pdf .