Using Semantic Networks for Knowledge Representation in an Intelligent Environment

Stephen Peters and Howard E. Shrobe

MIT Artificial Intelligence Laboratory, Cambridge, MA, 02139, USA,
{slp,hes}@ai.mit.edu,
http://www.ai.mit.edu/projects/iroom

Abstract. When building intelligent spaces, the knowledge representation for encapsulating rooms, users, groups, roles, and other information is a fundamental design question. Here we present a semantic network as such a representation, and demonstrate its utility as a basis for ongoing work.

1 Introduction

For many years now, research in intelligent spaces has grown, exploring different ways that a room can react to one or more users and their actions [1]. Much work has gone into defining new human interaction with these spaces, building systems for tracking users, and creating new uses for sophisticated learning and planning algorithms.

As use of these intelligent environments (IEs) grows, however, they will by necessity collect ever-increasing amounts of data about their users, in order to adapt to the user's desires. Information will be collected on the users' interests, who they communicate with, their location, web pages they visit, and numerous other details that we may not even notice. All this information needs to be collected and organized into a structure within the environment, so that the IE can make quick, correct assumptions about what the user would like to do next.

Here at the Intelligent Room project [2], we have begun to define one such knowledge representation (KR), using semantic networks as the basis for the representation. As this research continues, we are beginning to discover some inherent advantages with this approach:

- Adding new information into the system is highly straightforward, often as simple as adding in a new datum and providing an appropriate link.
- Changing information is a highly localized operation, rarely requiring major changes to widespread pieces of the representation. Similarly, deprecating information can often be done by either augmenting or replacing the existing links.
- Making inferences is also fast and easy, provided there are efficient interfaces for retrieving all links of a given type flowing into or out of a given node.

It is our belief that semantic networks are the most appropriate representation for capturing and encapsulating the vast amounts of information entering an intelligent environment. In this paper, we will make the case for the requirements that an IE imposes on a knowledge representation, and argue that semantic networks go a long way towards satisfying those requirements.

2 Related Work

There is a large body of literature related to developing contextual systems for intelligent environments. Dey, Abowd, and Selber have created a "Context Toolkit" for detecting a room's state and using input events to trigger changes to the context-sensitive applications [3]. This toolkit has enabled location-based applications that track sets of users as they enter and leave buildings and an assistant for conference attendees. Lauff's API for ubiquitous computing takes device inputs and sends signals to components. Martin, Cheyer, and Moran's Open Agent Architecture (OAA) [4] includes some facilities for triggering actions based on contextual information. Within our lab, Ajay Kulkarni has created a reactive behavior system called ReBa [5] which can trigger many different actions in response to device events. However, for all of these frameworks, the interpretation and inferencing is done by the applications, which must themselves collate and assemble disparate pieces of data, and little seems to be done in developing a cohesive representation of the knowledge gained.

Microsoft's EasyLiving project has done work on geometric representations of the space, and using that information to enhance the interactions of the user [6]. Although they pull a lot of information into this framework, there is no attempt to unify it into a grander vision of knowledge.

3 Knowledge Representations and Intelligent Environments

When discussing the kind of representation that is appropriate for these kinds of instrumented spaces, we can turn to the requirements that such a KR should provide. Davis et al. laid out five different roles that KRs play [7], and here we examine them within the context of an IE.

3.1 Surrogates for the Real World

First and foremost, the representation has to act as a computational surrogate for real-world entities. In the case of an intelligent space, many of the entities are easily identifiable, such as users, the spaces themselves, and devices within the space such as projectors, cameras, lights, appliances, computers, etc. Some are less easily defined, but still have real-world counterpoints, such as groups of people, roles that they play, actions that need to be performed, etc. These items need to be involved in the KR so that the space can reason about them and make deductions based on their layout.

One of the side effects is that the KR's surrogate for a real-world entity needs to closely track the actual entity's state, to avoid incorrect inferences. This can be greatly helped by using a KR which can easily be augmented with new information – including categories of information that were not planned when the system was designed.

3.2 Ontological Commitments

Just as the KR represents what is in the world, it also imposes a filter on what the environment can see and act on. For example, a knowledge representation for logic circuitry can view a circuit as a set of AND, OR, and NOT gates, and can thus examine the circuit's behavior based on the functions that those gates perform. However, that representation means that the same representation probably can't be used to, say, look for wires that have been shorted together, or examine the electron flow through the design. Conversely, a KR that operated at the lower levels likely would have trouble analyzing circuitry at the logic level. Davis et al. refers to the filter that the representation is applying as the set of *ontological commitments* implied by the representation.

For an IE, the commitments define how the IE looks at the world. When dealing with camera input, does it view people as such, or as the blobs of color that it is tracking through the room? Do we recognize speech input as belonging to a person or just as a waveform coming from a given microphone? The IE actually needs to operate at both these extrema (and perhaps a few in between), since the cameras will only be able to report on the blobs being tracked, but that will need to be associated with a person at some point. The KR, therefore, should be able to provide links between those levels.

3.3 Theories of Reasoning

The KR also defines how inferences are made by the reasoning system, and places bounds on what kinds of inferences are allowed or recommended by the system. For the most part, the IE does not impose extra constraints on this view of the representation. However, it does need to base many of its inferences on the space's current context, pulling in a large variety of data about the environment, including device states, currently running applications, users, presence of mobile devices, and a wide variety of other sensors and actuators. All this information contributes to the inferences that the IE makes about the world.

3.4 Media for Computation

Since the KR operates within a computational space, it must be able to operate in an efficient manner via computational processes, without taking up inordinate quantities of time, memory, or disk space. Due to the large amount of data that an IE must be able to process, these requirements are perhaps even more important. The KR must be able to collect large quantities of information together and efficiently link them together.

3.5 Media for Human Expression

Lastly, the KR is used to represent the human view of the world. As such, it requires that there be representative objects for each of the items that we describe in the environment, and that those representations function as an straightforward way of describing the world. Given that, the KR needs to be simple to use and to turn into human-readable representations. One of the outcomes of this requirement is that the addition of data should be fairly localized; when entering new information into the system it should require changes to only a few small, well-defined places in the representation.

4 Scenario

As an example of what a good knowledge representation for an IE should provide, consider the following scenario:

> Amy walks into her office and notices that she has received an email update. Reading the mail, she discovers that it is a synopsis of a meeting that took place earlier that day. Normally, she attends this meeting, but had to miss this one because of a doctor's appointment. However, the meeting management software was able to recognize both that she was normally a regular attendee, and that she was not present, so it generated the synopsis of the meeting discussions and sent it to her.
>
> The synopsis highlights the major discussions, and goes into more depth during the status reports from her staff. The synopsis also flags a few items of interest to her especially – most notably an action item that has been assigned to her.
>
> To get more information on that particular point than the synopsis can provide, she clicks on a link for that item and is able to browse the meeting record in more depth. From there, she can also call up a video-recording of the meeting, cued up to the time when the action item was first described.
>
> Since that particular item can be satisfied simply by providing some numbers from the latest budget report, she annotates the meeting record with the necessary information and some explanatory notes of her own. For good measure, she asks the computer to "fetch the group's budget report", and adds in a link to the resulting budget document. Afterwards, information on the annotations is then emailed to the members of the meeting involved in that discussion. In addition, the meeting management software will recognize that this item has been completed, and add it in as an agenda item for review by the team.

The above indicates the level of knowledge that users are likely to find useful. Having meeting spaces that track meeting attendees, recognizing and assisting people who can't be present, allowing people to work off-line, recognizing human relationships such as coworkers and supervisors – all of these are important to creating intelligent environments that assist us.

5 Semantic Network Description

In order to implement a scenario like the one above, and in keeping with the constraints that an IE imposes on the representation, we require a KR that can easily encapsulate the many different objects and associations implied by the text, including

- *people* such as Amy and the meeting attendees,
- *meetings* and any recording made,
- *documents* such as the budget,
- *discourse items* like the action item and the annotations Amy provides,
- *attendance* (so that the system can recognize that Amy did not attend a meeting for which she is usually present),
- *ownership* of action items,
- *responsibilities* such as Amy's responsibility for her staff,

and the list continues. What is needed is a representation that can store information on objects such as people and meetings, but also can easily track and follow the relationships between them.

One such representation is a "semantic network", which has its roots in Quillian's work on reasoning in computer environments [8]. This work represented knowledge as concept nodes related by directional relationship links, representing the world as a directed graph.

Exploring the framework of knowledge in such a system, therefore, is as straightforward as moving from a node along one or more links to discover related information. For example, finding all of Amy's co-workers might be as simple as starting at the node representing Amy, moving along an "in-charge-of-group" link to find the organization she runs, and then following the "member-of" links from that group to the individual workers within. This structure makes it easy for an intelligent system to uncover information about a particular topic, as well as to discover the relationships between two different objects.

The semantic network also satisfies many of the other conditions imposed by the IE. It is easy to encapsulate into human-readable form (either through graph networks or simple text representations); it's simple to add new information in a localized fashion; and making inferences is often merely collecting links and following them.

Much of the work behind semantic networks is being continued on a grander scale by the World Wide Web Consortium as part of their Semantic Web project [9]. Much of that project focuses on addressing the central problem of larger semantic networks – most notably, the problem of unifying large or multiple ontologies.

The problem with multiple ontologies is straightforward to describe, but difficult to overcome. Although it is easy to describe a mapping from a human-centric view of the world (people, places, things) into a set of descriptive names, such a mapping is bound to be highly domain- or location-specific. When different people try to create their own mappings, they either have to shoehorn their own mental model of the mapping into one created by somebody else, or create their own, separate mapping. The Semantic Web project has spawned technologies for translating one ontology into another, so that people can choose either to use someone else's ontology or "roll their own" as they see fit. One such technology is Fensel et al.'s Ontology Inference Layer (OIL) [10]. Although our current work is not focused on overcoming the ontology problem, we are looking towards OIL and related projects as extensions to our work.

6 Using a Semantic Network in an Intelligent Room

We have begun deploying this technology within the Intelligent Room project at MIT's Artificial Intelligence Laboratory [11, 2]. Our current implementation works on top of a SQL database storing the networks information, but can easily be extended to use object-oriented databases or even a Resource Description Framework (RDF) [12] back-end if desired. Software agents written in the Intelligent Room's Metaglue language [13] are able to create, query, and update the semantic network information as the user requires.

While developing this system, we have identified several areas in which using a semantic network-based representation is an appropriate and valuable piece of infrastructure for intelligent environments such as this one, most notably in the areas of user information, meeting management, and location infrastructure.

6.1 User Knowledge

One of the key pieces of knowledge for any intelligent space is that of the users and the individual spaces they work with. At a simplistic level, this can simply be a set of objects comprising spaces, how they are encapsulated within each other, and the user's current location (Figure 1). This gives access to straightforward queries like "where is Steve located" and allows for simple resource management dependent on the task and space involved [14].

However, in order to make a system that truly acts as an "intelligent assistant", you need to include far more information about the people and their relationships. Such systems need to be able to respond to queries and requests such as:

- "Send this information to everyone in the group."
- "Who is Joe's superior?"
- "Do I know the person who is responsible for this group's activities?"

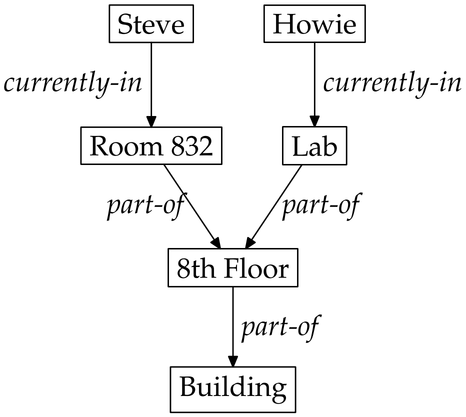


Fig. 1. Simple Layout for Users and Spaces

For this, we augment the network with information on groups of people, and the relationships between people, including notations about responsibilities and hierarchies (see Figure 2). In addition, we are providing information on interests and expertise to enhance the "intelligent assistant" role of the Intelligent Room. Such information will enable our agent systems to respond to more complex requests for information, and provide introductions to enhance communications between users:

- "Send this information to group members interested in HCI."
- "Do I know anyone who is an expert in writing LISP code?"
- "Who do I know who can introduce me to a LISP expert?"

We can also use such a system as part of the room's security model. Role-based access controls [15] are a reasonable model for limiting access in an intelligent room; for example, allowing all users the ability to turn on and off lights through voice commands, but only allowing users who are considered room administrators the ability to add new devices into the space. When using such a role-based access control (RBAC) mechanism, the semantic network is an extremely useful representation. A node for the "room administrator" is created, and links from people (or other groups of people) to this node are instantiated to indicate membership within the administrator role. A security subsystem can then follow the membership links for a user to find out what roles (and therefore what access) the user maintains.

Using the semantic network for security models can also provide for more interesting, situation-dependent control that isn't possible with simpler RBAC setups. For example, even if all users can control the room's devices most of the time, it may be that special access is set up for meetings – for instance, allowing only the current meeting's facilitator to grant access to the room's devices for presenting information.

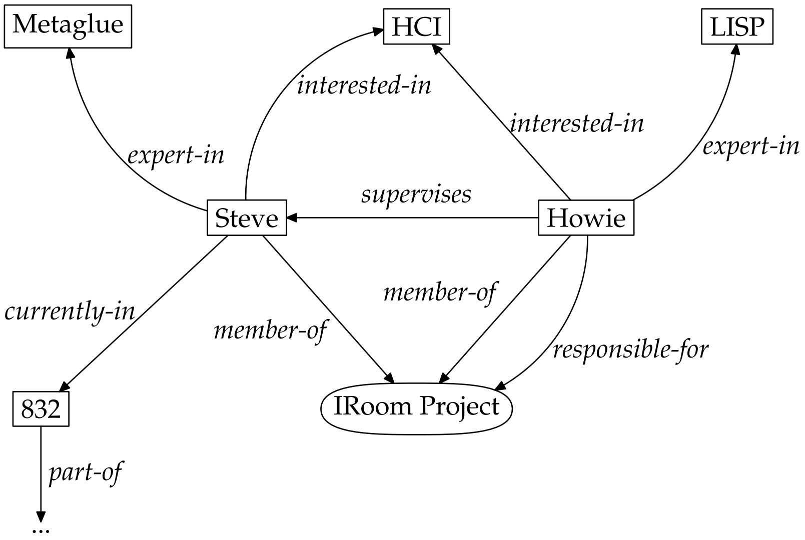


Fig. 2. Extended layout, adding information on interests, expertise, hierarchies, and groups

6.2 Meeting Management

We are now using the semantic network structures to capture meetings as they occur, linking together the main meeting topics along with their contributors and attendees. Typical information that gets captured during a meeting includes agenda topics, action items, supporting and dissenting arguments, and documents such as presentations or web references. People are linked in as meeting attendees, document authors, and as issue-raisers. When meetings take place in instrumented environments, they can be linked to a video or audio capture of the meeting in progress.

Using the philosophy that meetings are not the primary piece of information, but merely a framework for examining and disseminating information, the discussion topics within the meeting management software can be linked together. This makes it possible to review proposals as they travel through a long-term set of meetings, and to ask the system questions regarding previous meetings discussing the current topic.

6.3 Location Management

Another important piece of information that IEs need to deal with is location information. This is especially true when you integrate hand-held devices into the environment. Such devices are extremely useful tools both for navigating around spaces as well as for sharing information with others [16], but require knowledge of the current location, provided either through local IR- or camera-based tracking systems, or through more widespread systems like radio networks or GPS.

One problem with this is that all these systems use their own coordinate framework to describe a person's location, and provide their own resolution for coordinates. In addition, queries about location have their own granularity as well: asking "Where is Amy?" should probably have a different answer if the task is deciding which device she is trying to address, as opposed to when the task is deciding whether to send an email message or talk to her pager.

We are using the semantic network to assemble a location infrastructure for the Intelligent Room project. This will allow for the integration and translation between the different coordinate systems, as well as allowing users to specify their own location descriptions, such as "at home."

7 The Meeting Management Application

It is useful to examine one of these applications in depth to get a real sense of how the network gets created and what it can provide. For this reason, we will go into some detail here about how our MeetingManager application utilizes the tools of the semantic network and the Intelligent Room.

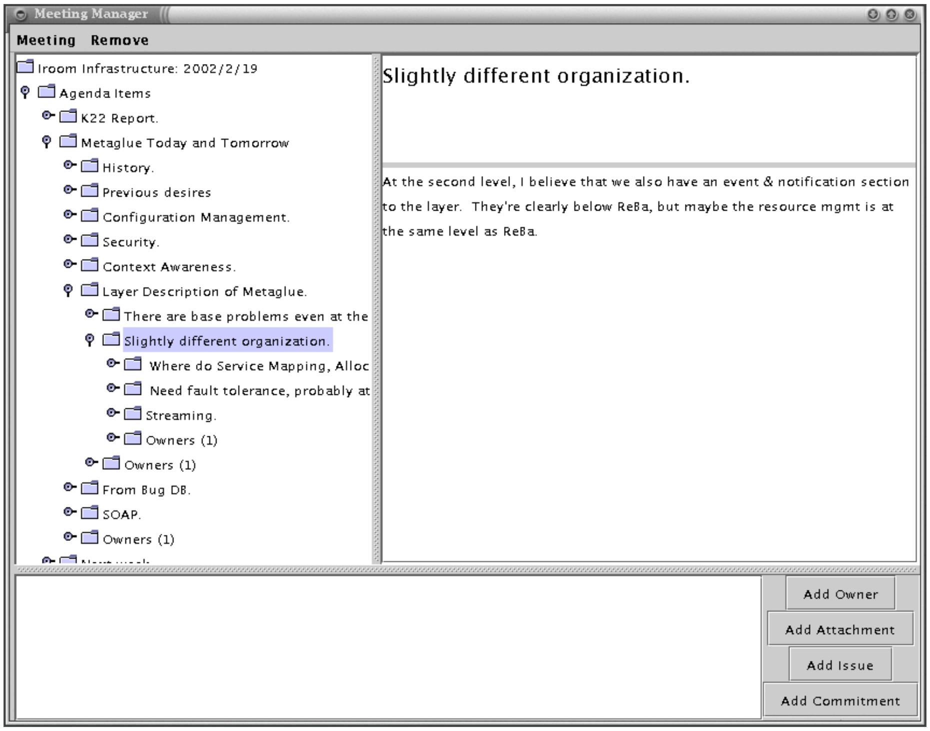
The Intelligent Room project uses the Java-based Metaglue agent infrastructure [13] to build agents that can communicate with each other. These agents are identified by their function, so that, for example, the agent that activates projection screens within a room is called the ProjectionScreen agent. For the most part, agents communicate with each other through direct, one-to-one remote method calls, although broadcasting facilities are available.

The current version of the MeetingManager application builds upon initial work by Oh et al. [17], who created an application to demonstrate the use of an intelligent room in a collaborative meeting context. That application maintained information about the agenda, major issues raised, and any commitments agreed to during the meeting time. It could also record information to link in these events with a QuickTime recording of the session. Although it performed well as a demonstration, it was hampered by a lack of robustness in the data model, with an inability to capture anything deeper than the broadest points of a discussion, and no capacity for reviewing and augmenting the discussion off-line.

This evolution of the MeetingManager application encompasses several agents that act in concert. The first of these, MeetingModel, serves merely as an interface to the semantic network and abstracts out some of the network lookup tasks into simpler methods. It also has the ability to broadcast any changes made to the meeting structures to other agents that request them. We suspect that this will likely be a design feature of many of the Intelligent Room's semantic network applications, since having one agent which can coordinate and monitor the KR for appropriate changes prevents duplication of code and provides for better data abstraction.

Other agents are available which provide different interfaces to the meeting manager data. One, the Gui agent (see Figure 3), builds a tree-structure view of the meeting information and allows easy creation and editing of topics, issues, and discussion points. This editor serves fairly well as a primary interface for a meeting recorder as he or she takes notes. Currently, this is the primary conduit for meeting information to get into the semantic network, although we are hoping to have the Intelligent Room add in some of this information itself (such as automatically recognizing meeting attendees and using knowledge about seating arrangements to determine who is raising the discussion points), and to provide more robust interaction with the room (such as using more voice commands to create nodes).

Other viewing and editing agents provide different views of the data – through web interfaces, graph networks, or specialized meeting views which simply present the agenda items or commitments. All of these are updated as information gets added to the semantic network. Meeting attendees can decide to bring up their own person view of the meeting state, so that they can augment and add the meeting information with more information than the meeting recorder was able to provide.



- Iroom Infrastructure: 2002/2/19
  - Agenda Items
    - K22 Report.
    - Metaglue Today and Tomorrow
      - History.
      - Previous desires
      - Configuration Management.
      - Security.
      - Context Awareness.
      - Layer Description of Metaglue.
        - There are base problems even at the
        - Slightly different organization.
          - Where do Service Mapping, Alloc
          - Need fault tolerance, probably at
          - Streaming.
          - Owners (1)
        - Owners (1)
      - From Bug DB.
      - SOAP.
      - Owners (1)

Slightly different organization.

At the second level, I believe that we also have an event & notification section to the layer. They're clearly below ReBa, but maybe the resource mgmt is at the same level as ReBa.

Fig. 3. The MeetingManager's Gui.

The network itself stores a variety of node and link types for the meeting, including meetings, multimedia recordings, documents, people, issues, commitments, and other discourse items. Most of these types actually have very little information attached to each of them, as the semantic network relies more on the links between nodes to define the relationships. For example,

– a person can act as an *author* of a document, an *attendee* of a meeting, a *facilitator* for a meeting, or an *owner* (sometimes considered the "raiser") of a discourse item;
– discussion points can be raised as a *supporting* argument, a *dissenting* argument, or even as an *implication* to a preceding discussion;
– discourse items can be marked as *agenda* nodes so that they can be used to organize meeting flow.

The endpoints for the links are not limited to the storage nodes; links can also be created to other links. This allows more complex interactions in the data model, so that a person can augment a supporting argument with a node describing his own agreement. We use this ability to augment links extensively to register that a link was created during a meeting, so that we can later request information on any discussion points raised at a certain time, and link those points to the meeting video.

With many discussion points being raised during a meeting, these networks can grow to be fairly complex, as seen in Figures 4 and 5.

When the meeting is over, the network makes it possible to use voice commands to delve into the structure and answer many questions about the meeting, among them:

– "What members of the Intelligent Room Project group did not attend the meeting?"
– "What open commitments are assigned to me?"
– "What issues were raised in opposition to this discussion point?"
– "What points did Krzysztof raise?"
– "Show me the video for this discussion point."

The last example works by coordinating the timestamp for the discussion points with that of the meeting video, so that it can skip forward to the appropriate position during the video playback. By linking together multiple meetings, it is also possible to make these queries about the meeting history, and retrieve the archived footage from previous discussions, allowing for a quick recap of the important events.

8 Conclusions and Future Work

As we have found, the semantic network buys us a great deal of flexibility in terms of the data we capture and interpret. Because of the nature of the semantic networks, adding and changing information is localized and clear-cut, and inference generation can be done extremely efficiently.

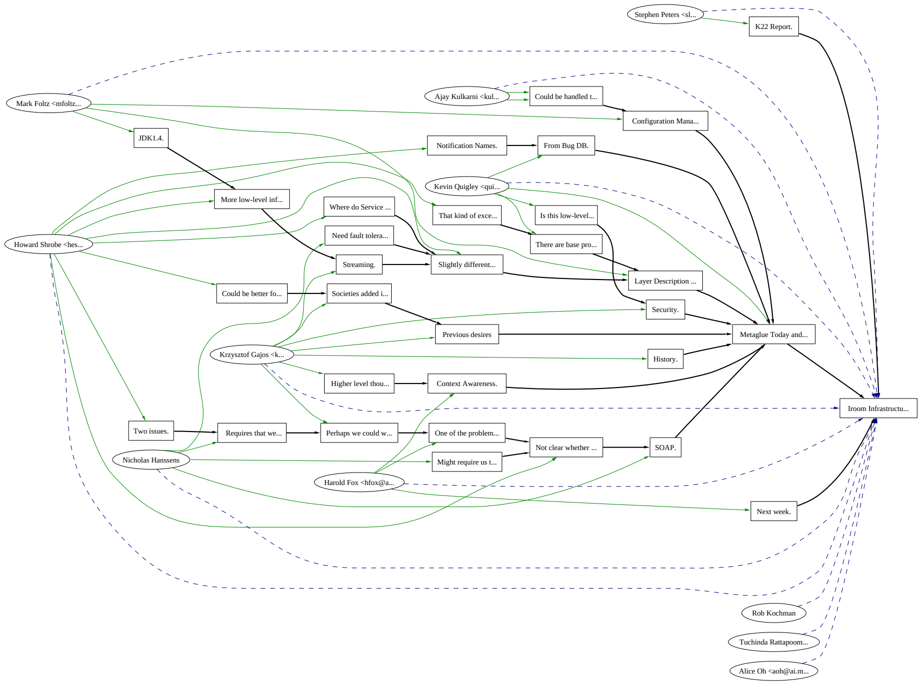


Fig. 4. The complexity of a discourse structure captured during a typical "brainstorming" meeting. People are represented by ovals, discourse items by rectangles. Different relationships are represented in different colors. A closeup of this is in Figure 5.

We are exploring different ways to enrich the interactions we can provide to the user. As one example, we would like to introduce a network activation system into the network's searching capabilities. This would allow a user (or an agent operating on the user's behalf) to specify a segment of network that it is looking for, and get notified whenever that piece of network is created. For example, a user might request to be notified whenever a topic of interest to him is raised during one of the room's meetings. The user's agents would then translate that request into an associated fragment of the semantic network, as shown in Figure 6.

Naturally, being able to determine what topics a user finds interesting is itself an impressive task. Our current plan is to allow the user to define his or her own interests directly, or to determine it automatically by integrating pieces of the system with the Haystack

Further work will likely include integration with the Semantic Web system and the associated frameworks for handling multiple ontologies, as noted previously. This may also include integrating in our knowledge framework with the Haystack project at the MIT Laboratory for Computer Science [18], so that we can better define interests and information retrieval.

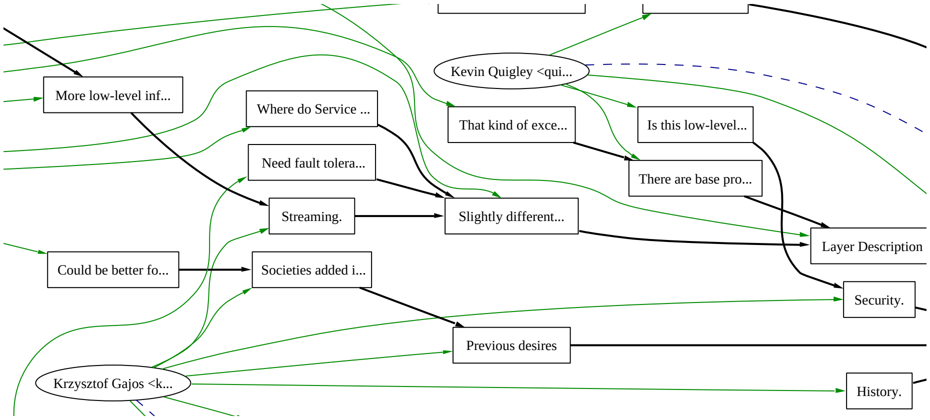


Fig. 5. Closeup of Figure 4. The links from users to discourse items specify the "owner" of each discussion point, whereas links between discourse items are discourse links.

```
(define notify-on-meeting-interests
   (and (type-of ?n1 Meeting)
     (during ?n1 ?argument)
     (topic ?argument ?top)
     (interested ?top ?me))
   =>
   (send-message-to ?argument ?me))
```

Fig. 6. A trigger for the semantic network (using a CLIPS-like syntax)

References

[1] Michael Coen, editor. *Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments.*, Menlo Park, California, 1998. American Association for Artificial Intelligence, AAAI Press.

[2] Nicholas Hanssens, Ajay Kulkarni, Rattapoom Tuchinda, and Tyler Horton. Building agent-based intelligent workspaces. In *ABA Conference Proceedings*, July 2002. To appear.

[3] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A context-based infrastructure for smart environments. In Nixon et al. [19], pages 114–128.

[4] D. Martin, A. Cheyer, and D. Moran. The Open Agent Architecture: a framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1/2):91–128, 1999.

[5] Ajay Kulkarni. Design principles of a reactive behavioral system for the Intelligent Room. *Bitstream: The MIT Journal of EECS Student Research*, 2002. To appear.

[6] Barry L. Brumitt and Steven Shafer. Better living through geometry. *Journal for Ubiquitous Computing*, 5, January 2001.

[7] Randall Davis, Howard E. Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.

[8] M.R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Cambridge, MA, 1968.

[9] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.

[10] Dieter Fensel, Ian Horrocks, Frank van Harmelen, Stefan Decker, Michael Erdmann, and Michel C. A. Klein. OIL in a nutshell. In *Knowledge Acquisition, Modeling and Management*, pages 1–16, 2000.

[11] Michael Coen. The future of human-computer interaction or how I learned to stop worrying and love my Intelligent Room. *IEEE Intelligent Systems*, (March/April), March 1999.

[12] Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. W3C recommendation. Technical report, World Wide Web Consortium, 1999.

[13] Michael Coen, Brenton Phillips, Nimrod Warshawsky, Luke Weisman, Stephen Peters, Krzysztof Gajos, and Peter Finin. Meeting the computational needs of intelligent environments: The Metaglue system. In Nixon et al. [19], pages 201–212.

[14] Krzysztof Gajos. Rascal - a resource manager for multi agent systems in smart spaces. In *Proceedings of CEEMAS 2001*, 2001.

[15] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[16] Jun Rekimoto. A multiple device approach for supporting whiteboard-based interactions. In *Proceedings of the CHI 98 Conference on Human Factors in Computing Systems*, pages 344–351, Los Angeles, California, USA, 1998. ACM.

[17] Alice Oh, Rattapoom Tuchinda, and Lin Wu. MeetingManager: A collaborative tool in the Intelligent Room. In *Student Oxygen Workshop*, Cambridge, MA, 2001. http://www.ai.mit.edu/projects/iroom/publications/alice-sow01.pdf.

[18] E. Adar, D. Karger, and L. Stein. Haystack: Per-user information environments. In *Proceedings of the 1999 Conference on Information and Knowledge Management, CIKM*, 1999.

[19] Paddy Nixon, Gerard Lacey, and Simon Dobson, editors. *Managing Interactions in Smart Environments: 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99)*. Springer-Verlag, London, 1999.