Software Agents in Support of Human Argument Mapping

Simon BUCKINGHAM SHUM[a], Maarten SIERHUIS[b,c,d], Jack PARK[a], Matthew BROWN[e]
[a] *Knowledge Media Institute, The Open University, UK*
[b] *NASA Ames Research Center, USA*
[c] *Carnegie Mellon University, Silicon Valley, USA*
[d] *Man-Machine Interaction, Delft University of Technology, NL*
[e] *University of Utah, Salt Lake City, USA*

Abstract. This paper reports progress in realizing human-agent argumentation, which we argue will be part of future Computer-Supported Collaborative Argumentation (CSCA) tools. With a particular interest in argument mapping, we present two investigations demonstrating how a particular agent-oriented language and architecture can augment CSCA: (i) the use of the *IBIS* formalism enabling *Brahms* agents to simulate argumentation, and (ii) the extension of the *Compendium* tool by integrating it with Brahms agents tasked with detecting related discourse elsewhere.

Keywords. Argument Mapping, IBIS, Compendium, Brahms, Multi-Agent Systems

1. Introduction

Computer-Supported Collaborative Argumentation (CSCA) research seeks to augment human dialogue, deliberation and argumentation with appropriate software support, with a significant interest in how the visualization of these discourse structures can augment personal and shared cognition [1,2,3]. As part of a long-term research programme, we have been developing an open source and open architecture tool for CSCA mapping called Compendium [4]. In tandem with experimenting with the technological possibilities (e.g. through integration with video-conferencing, social media, or AI planning [4]), we are studying its use in authentic contexts, in order to determine the work practices that make such tools effective [5,6].

Argumentation in Multi-Agent Systems (MAS) research is investigating how argumentation theory can provide software agents with greater capacity to reason and negotiate, in order to resolve competing priorities or recommended courses of action. Parallel to this, our long-term objective is the design of human-agent interaction (HAI), a subset of which implicates human-agent argumentation. In this paper, we report on two investigations into how Brahms, a particular agent-oriented language and architecture developed at NASA [7] can augment CSCA. The efforts we report on cover (i) the use of the Issue-Based Information System (IBIS) formalism [8] enabling software agents to interact about competing options using argumentation, and (ii) the extension of the Compendium CSCA tool by integrating it with agents tasked with detecting related discourse elsewhere and enabling the ability to share discussion topics and argumentation between different context. To convey the kind of use case that we are designing for, consider this scenario:

> *Susan is one of a team of analysts at a commodities trading firm. The team is seeking to understand how climate change may impact the markets they follow, and have assigned different staff to specific topics. Aware that this is a highly contested issue, Susan creates a new project in her argument mapping tool, and records the results of her research in a set of maps, tracking Questions, potential Answers, and relevant Arguments to help reflect on the trustworthiness of the information she is gathering. She skims a climatology book to identify major themes, before going online to examine government advice, business analyses, and climate change advocacy groups from the different sides. Her argument-mapping tool includes a supporting personal software agent that in the background has access to a network of other intelligent software agents that can check if there are related conversations/analyses on the net. She can view suggestions from her agent, and add the most relevant to her own analysis. Some of the suggestions come from agents monitoring her colleagues' maps; others are mining online discussions in interoperable platforms, while others are trained to perform their own research on specific online databases. Moreover, some agents are capable of working together to construct their own argument maps around a given issue, which are then proposed to the analyst as contributions.*

In order to examine the modelling and implementation requirements raised by this, we start (§2) by introducing our previous work in CSCA, in particular our development of the Compendium software tool to support IBIS and other notations. We then introduce the Brahms agent language, and summarise progress to date on human-agent argument mapping from a series of NASA field trials (§3). This sets the context for a deeper level of integration, which serves as the core of this paper, namely the extension of Brahms agents to conduct IBIS-based analyses (§4), and the extension of Compendium with Brahms agents to identify potentially relevant IBIS content in remote databases (§5). We then draw conclusions and consider future work.

2. Our Previous Work in CSCA

In [9] we trace the work of design and policy planning theorist Horst Rittel, whose characterisation in the 1970's of "wicked problems" continues to resonate with today's societal challenges: "Wicked and incorrigible [problems]... defy efforts to delineate their boundaries and to identify their causes, and thus to expose their problematic nature" [10]. In such domains, the complexity of the arguments invites CSCA support, although as we discuss elsewhere [11], the evidence is that before deep dialogue has helped to build common ground and trust, clear argumentation on its own will never be sufficient. Rittel concluded that real world policy dilemmas are qualitatively different to those that could be solved by formal models or methodologies, classed as the 'first-generation' design methodologies. Instead, an *argumentative* approach to such problems was required: "First generation methods seem to start once all the truly difficult questions have been dealt with. … [Argumentative design] means that the statements are systematically challenged in order to expose them to the viewpoints of the different sides, and the structure of the process becomes one of alternating steps on the micro-level; that means the generation of solution specifications towards end statements, and subjecting them to discussion of their pros and cons.” [10]

Rittel’s work has proven influential in CSCA research, through his proposal of the Issue-Based Information System (IBIS) as a method and notation for conducting “argumentative design”. The gIBIS prototype [12] and the subsequent QuestMap product, rendered IBIS as a graphical-hypertext network. A root *Issue* provides the orientation to a map, establishing the problematic context for the discussion. The analyst then maps possible responses to these, and relevant arguments. An important strand of our work has sought to articulate the skillset that practitioners deploy when they use such tools to add value to Design Rationale capture, and other forms of knowledge-intensive work [6].

Several discourse-modelling methodologies have developed around the capabilities of Compendium. *Dialogue Mapping* is a set of skills developed by Conklin [13] for mapping IBIS structures in real time during a meeting in order to support the analysis of wicked problems, as defined by Rittel. *Issue Mapping* is conducted asynchronously, without the pressure of real time knowledge representation, permitting more reflection prior to crafting the map, which is more typical of argument diagramming tools. In Dialogue and Issue Mapping, nodes are usually unconstrained free-text expressions summarising an agenda item or a participant’s contribution (Figure 1). “Arguments” are typically no more formally expressed than as shown in the examples below, because the demands of real time mapping do not permit greater analysis and formalization.

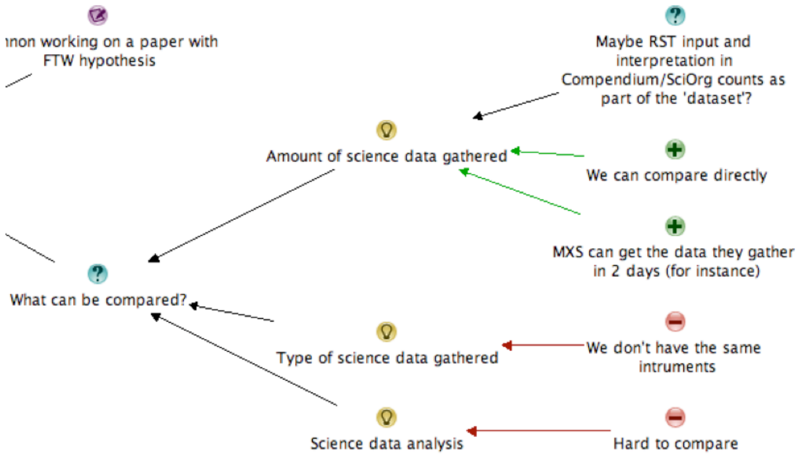

Figure 1: Example of an IBIS map constructed in an online meeting, shared live over the internet

The more disciplined *Conversational Modelling* technique [14,15] incorporates and extends Dialogue Mapping by using a modelling methodology as the driver for the particular kinds of Issues, Answers and Pros/Cons that are mapped. *Templates* are defined to seed particular genres of analysis, optionally constraining node labels to machine-readable strings, and using node typing and tags to assist automation. Conversational Modelling templates have particular relevance for argument modelling in general, and for this paper in particular. Firstly, we have reported how Walton’s critical questions associated with different presumptive argumentation schemes, once published in XML by Reed and Walton, were then further transformed into IBIS Conversational Modelling templates in Compendium [11]. For instance, a *challenges* link, which is in fact making an *argument by analogy*, can be ‘exploded’ into a template map showing the implicit premises, and the associated critical questions that can be asked about them.

Secondly, in the context of two space exploration field trials, we have reported how templates enabled software agents both to read and write Compendium maps [16], enabling Compendium to play a multiplicity of roles:

- As a way to create formal information structures for understanding by software agents, from informal discussions by people.
- As a way to navigate richly linked data and metadata in maps written by software agents;
- As a real time discourse mapping environment for both co-located and online meetings;
- As an asynchronous medium for distributed team conversation;
- As an asynchronous medium for scientists to program software agents;
- Combining planned, formal modelling, with interpretive scientific and project management discourse which could move in unpredictable directions.

Compendium is implemented as an open source, cross-platform, Java application that can swap between either the MySQL or Apache Derby relational databases. SQL and XML export/import assists data interoperability between clients and servers, and a number of projects have used RDF for data interchange. Public Java application interface classes provide an interface for other systems to read and write to the database directly, so maps can be generated from another data source or interpreted for processing by another system. A shared MySQL database supports group working over a local area network or Internet. Maps can be published for web browsers to view as interactive image-maps, or as linear HTML outline documents. An active user and developer community are supported by the Compendium Institute, which has logged over 30,000 downloads of the tool to date [17].

Recent years have seen the emergence of other web platforms for structured deliberation, whose core ontology is IBIS. Debategraph [18], Deliberatorium [19], and Cohere [20] seek to exploit the benefits of a web user interface and Web 2.0 services. An ongoing project is designing and testing a common serialization format for these platforms [21]. Apart from the possibility for diverse platforms to exchange data, web services and agent architectures point to the possibility of federating IBIS dialogue/argumentation [22].

This represents the current state of our agent-augmented CSCA research, and computational platform, and serves as the point of departure for the new work. The purpose of the work presented here was to show the feasibility of our approach, in order to ultimately develop an agent system that enables an agent-supported discussion, enabling software agents to negotiate with each other and/or people. Our prototype systems presented in sections 4 and 5 are a step towards showing that software agents can both simulate and support a human discussion using the IBIS argumentation framework. The idea is that if software agents can use IBIS, then it will be possible to enable human-agent discussions in a specific domain. These presented examples are a first step towards proving this hypothesis and ultimately developing a MAS system that can do this.

3. An Agent Language for Supporting CSCA

Brahms [7] is an agent-oriented language and agent simulation and execution environment that has been under development for over a decade. It provides a way to model activity and communication practices of groups of people and multi-agent systems, and contrasts to other belief-desire-intention (BDI) agent languages, as discussed in [23].

Brahms is a mature agent architecture used at NASA's International Space Station Mission Control Center [24], and underpinning large scale e-science infrastructures such as the Mobile Agents Architecture (MAA), which provides a means for modelling, simulating, implementing and managing a computer-supported Mars/Earth-based science work system [25].

Brahms agents are belief-based and use situation-action rules to perform activities. One type of activity is communicating with other agents, whether other Brahms agents or agents implemented in the Java language. The Brahms language provides a communications library enabling the design of specific agent communication protocols, based on speech act theory [26], as defined by the FIPA agent communication standard [27]. Using a combination of belief-based Brahms agents and a standardized communications protocol for IBIS conversations between agents, we developed our Brahms framework for collaborative argumentation (CA). It is our belief that this CA framework can be extended to become a general framework for human-agent CA. Here we explain the Brahms communication library. In the next section we discuss the Brahms IBIS protocol.

The Brahms communication library defines the FIPA communication act as the Brahms object class *CommunicationAct.* An instance (called a *comact*) of this class has to be created by the agent that wants to communicate about a particular topic (called the *payload* of the comact). The other attribute of a comact is the comact's *envelope*, specifying the sender and receiver information. The Brahms *Communicator* group defines a number of activities for creating comacts and communicating (i.e. sending) comacts to other agents (a Brahms activity is a predefined action or plan taking an amount of time to complete). The payload content of a comact is not defined in the CommunicativeAct class, but is specified by FIPA and can be designed by the modeler.

4. Extending Brahms Agents to Conduct IBIS Conversations

Perhaps the biggest change in transitioning from human to agent IBIS argumentation is the role of the facilitator. For human IBIS argumentation, the facilitator is responsible for both translating specific comments made by the group into individual IBIS nodes as well as maintaining the overall structure of the IBIS conversation. With agent IBIS argumentation, there is no longer a need for the facilitator to parse suggestions and ideas from the participants into their proper IBIS notation. Instead, we can develop agents that are capable of expressing their beliefs directly through IBIS nodes. While the agent facilitator is no longer required to perform translation, the task of maintaining the structure of an IBIS conversation between agents is much more difficult than in the human case. It is easy enough for a human facilitator to identify if an idea being proposed has already been summarized in the IBIS conversation. Agents lack this innate ability and thus the agent facilitator must search through the IBIS conversation to ensure that an IBIS node is unique before adding it to the conversation. Without intervention, duplicate IBIS nodes would, at a minimum, increase the amount of computation without adding any semantic value, and in the worst case could result in infinite loops.

At an abstract level, the process of creating an agent capable of understanding and communicating in IBIS is straightforward. In order to understand IBIS, an agent has to be aware of the various types of IBIS nodes and the function that they serve. The type of IBIS node received, the beliefs encapsulated by the IBIS node, and the current beliefs of the agent determine how an agent responds to a given IBIS node.

4.1. Implementing IBIS-agents in Brahms

The IBIS framework is defined by two interfaces, IBISParticipantAgent and IBISFacilitatorAgent. These interfaces present the functions, which must be implemented with domain knowledge in order to conduct IBIS argumentation.

`IBISParticipantAgent`

- *`preArgumentationActivity()`*
  - defines the actions taken by an agent before the argumentation begins, this may include the sending of the initial IBIS nodes that start the argumentation
- *`postArgumentationActivity()`*
  - defines the actions taken by an agent after the argumentation has concluded, this may include deciding the outcome of the argumentation
- *`processQuestionNode(IBISNode node), processIdeaNode(IBISNode node), processProNode(IBISNode node), processConNode(IBISNode node)`*
  - defines the actions taken by an agent when processing the various types of IBIS nodes, this may include the creation of new beliefs and/or responding with an IBIS node

`IBISFacilitatorAgent`

- *`checkForDuplicate(IBISNode node)`*
  - defines the process by which IBIS nodes are determined to be unique or duplicate

Each IBISParticipantAgent starts its execution by calling *preArgumentationActivity()*. Then for a predefined amount of time the IBISParticipantAgents will periodically check with the IBISFacilitatorAgent for new IBIS nodes. Each IBISParticipantAgent will process new IBIS nodes by calling the appropriate function for that type of IBIS node to see if it matches any of the preconditions for generating a response. After time has expired, each IBISParticipantAgent calls *postArgumentationActivity()* and then quits. The IBISFacilitatorAgent waits to receive an IBIS node from an IBISParticipantAgent. When it does, it calls *checkForDuplicate()* on the incoming node to see if it has already received an identical node. If it hasn't, then the node is added to the IBIS conversation.

IBIS nodes and IBIS notation play an enhanced role in agent IBIS argumentation. In human IBIS argumentation, the IBIS notation is not the same as the discussion, but the IBIS notation is used as a captured summary of the real-life dialogue, serving to focus and improve the quality of the discussion and create shared understanding between the discussion participants. Contrast this with the agent IBIS argumentation case, where the actual "discussion"[1] between agents is taking place directly through the IBIS notation. Therefore, it is necessary to give the IBIS nodes for agent argumentation additional expressive power that is not present in human IBIS argumentation. IBIS nodes are represented in the framework as shown in Figure 2.

The most important aspect of an IBIS node is the *content* attribute. When an agent goes to process a new IBIS node, they can determine what beliefs are associated with this node by looking at the objects referenced in *content*. A problem arises when multiple IBIS nodes refer to a single object, as it is no longer possible to determine which beliefs should be associated with which node. This problem is circumvented by having each agent create a new copy of the object they want to talk about and reference the copy in *content*, which ensures that each IBIS node refers to unique objects.

---

[1] The word discussion is purposefully in between quotation marks, signifying the realization that agents are not having actual discussions, but are simply sending IBIS-like messages to each other, which are then interpreted in agent specific ways to partake in individual agent activity outside of the IBIS communication activity being held.

```
class IBISNode extends SerializableObject{
  attributes:
    int weight; //the weight of the IBIS node when deciding the outcome of the
                 argumentation
    IBISParticipantAgent sender; //sender of the IBIS node
    IBISNode parentNode; //IBIS node that this IBIS node is responding to
    map content; //map that contains references to all objects that appear in
                  the beliefs associated with this node
    string label; //text description of IBIS node (display use only)
    boolean isDuplicate; //indicates if an identical IBIS node has been seen
                          before
  relations:
    IBISNode childNode; //IBIS nodes that respond to this IBIS node
}
```

Figure 2: Representing IBIS nodes in the Brahms framework

4.2. Brahms IBIS-agents: preliminary evaluation

Simulating the Federal Aviation Administration's (FAA) Collaborative Convective Forecast Product (CCFP) chat sessions tested our agent IBIS argumentation framework. This online, text-based chat consists of meteorologists representing a variety of organizations (Aviation Weather Center, regional Air Route Traffic Control Centers, airlines, etc.) collaborating to form a consensus weather forecast. The forecaster for the Aviation Weather Center (AWC) is responsible for leading the chat as well as producing the final forecast. A typical CCFP chat begins with the AWC forecaster presenting their initial forecast. They then open up the discussion to see if any participants have modifications to suggest. The final forecast is then used as the primary weather analysis for the FAA's strategic planning teleconference.

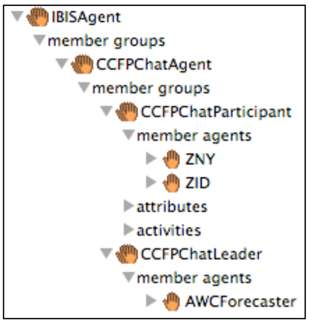

Figure 3. Brahms group membership for IBIS Framework

The text-based nature of the chat sessions appears to have a negative impact on the effectiveness of the collaboration. Problems include participants responding to questions out of order, failing to identify the question to which they are responding, and suggesting ideas without any kind of justification. As a result, the chat transcripts become convoluted, which diminishes their value as a record of the participants' decision-making process and serves to decrease the overall amount of participation. For these reasons, rather than simulating the CCFP chat sessions verbatim, we are suggesting the use of IBIS as a method for structuring the discussion.

The model we describe next is a first prototype of a scenario in which some of the knowledge held currently by human participants can be held and negotiated by agents. The model shows that the human collaboration during the CCFP chat can be formalized and modeled in an agent-based simulation, using the IBIS formalism.

Figure 3 shows that the *IBISAgent* interface is implemented by *CCFPChatAgent*. *CCFPChatAgent* is further extended by *CCFPChatLeader* and *CCFPChatParticipant*. The *AWC forecaster* agent is a member of the group *CCFPChatLeader*, where as all other participant agents are members of *CCFPChatParticipant*. The flexibility of the framework allows you to define multiple classes of agents that are capable of participating in IBIS argumentation. This could range from conducting IBIS argumentation with entirely homogeneous participants to having unique behaviors defined for each participant. Figure 4 specifies the beliefs held by the three IBIS agents in Figure 3.

```
agent AWCForecaster memberof CCFPChatLeader{
       initial_beliefs:
          (current updatedWeatherEvents unknown);
          (current.weatherReport = AWCForecasterWeatherReport);
          (AWCForecasterWeatherReport twoHourForecast WeatherEvent1);
          (WeatherEvent1.name = "weather_event_1");
          (WeatherEvent1.confidence = 1);
          (WeatherEvent1.growth = 1);
          (WeatherEvent1.tops = 1);
          (WeatherEvent1.coverage = 1);
          (WeatherEvent1.speed = 25);
          (WeatherEvent1.direction = 45);
}
agent ZID memberof CCFPChatParticipant{
       initial_beliefs:
          (current updatedWeatherEvents unknown);
          (current.weatherReport = ZIDWeatherReport);
          (ZIDWeatherReport twoHourForecast WeatherEvent1);
          (WeatherEvent1.name = "weather_event_1");
          (WeatherEvent1.confidence = 1);
          (WeatherEvent1.growth = 1);
          (WeatherEvent1.tops = 2);
          (WeatherEvent1.coverage = 2);
          (WeatherEvent1.speed = 25);
          (WeatherEvent1.direction = 45);
}
agent ZNY memberof CCFPChatParticipant{
       initial_beliefs:
          (current updatedWeatherEvents unknown);
          (current.weatherReport = ZNYWeatherReport);
          (ZNYWeatherReport twoHourForecast WeatherEvent1);
          (WeatherEvent1.name = "weather_event_1");
          (WeatherEvent1.confidence = 2);
          (WeatherEvent1.growth = 2);
          (WeatherEvent1.tops = 2);
          (WeatherEvent1.coverage = 1);
          (WeatherEvent1.speed = 25);
          (WeatherEvent1.direction = 45);
}
```

Figure 4: Three Brahms IBIS agents with beliefs derived from the CCFP use case

When we simulate the CCFP chat session conducted by running these agents in the Brahms environment, an IBIS conversation is produced: Figure 6 shows the Brahms simulation output, and Figure 7 shows the visualization of their IBIS argumentation, an initial result that gives us confidence that this approach has potential. However, Figure 5 shows the exponential rate at which simulation time increases with the number of IBIS nodes generated. Work needs to be done to bring this rate closer to linear before the framework will scale. The performance issues are due to the fact that an agent is required to create a copy of each object that it references in an IBIS node before communicating it to the facilitator. Finding a solution to this issue should help reduce the memory footprint of the framework, which in turn should increase the overall performance. Another issue facing the IBIS framework is the difference in complexity of the code for the framework, compared to that of the domain specific implementation. While the framework itself is simple and straightforward, the domain specific implementation is convoluted and repetitive. Making the framework more sophisticated should lead to a reduction in the amount of both code and effort necessary to apply it to new domains.

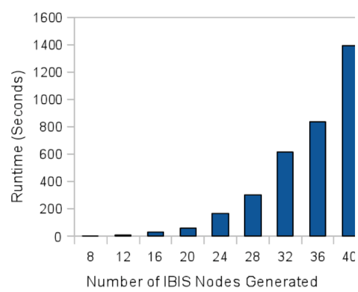

Figure 5: Runtimes for simulated CCFP IBIS conversations

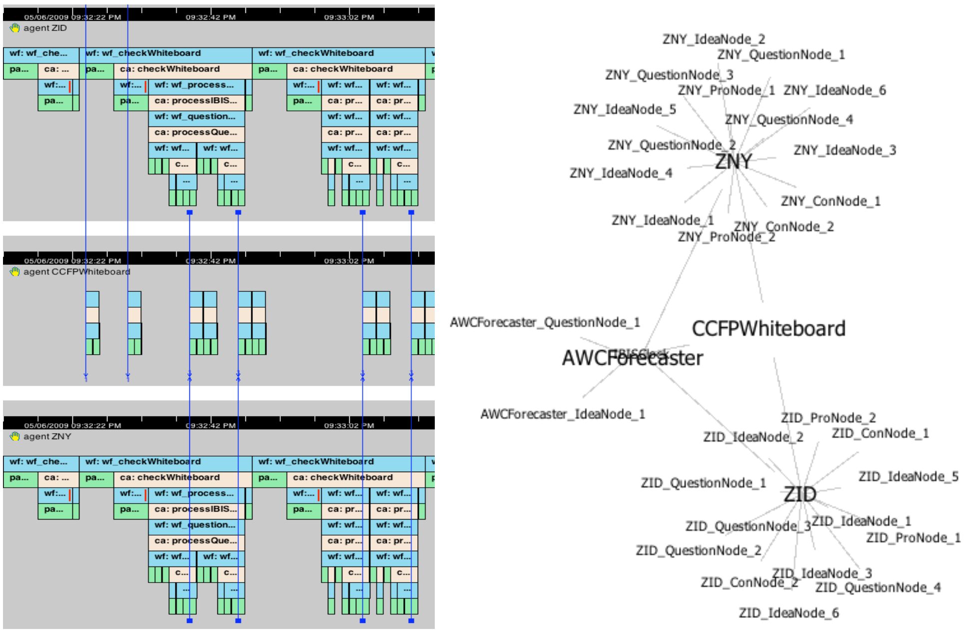

Figure 6. Brahms simulation output visualization of simulated CCFP IBIS conversation

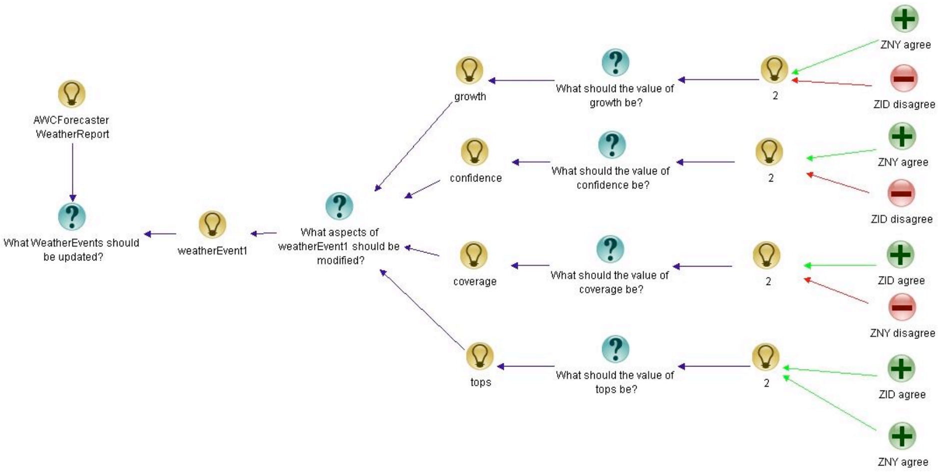

Figure 7: Compendium visualization of simulated CCFP IBIS conversation

5. Extending Compendium with Brahms Agents

We turn now to the second investigation into human-agent argument mapping: extending our CSCA mapping tool (Compendium) with agents (Brahms), in order to enable agent-based search of remote IBIS maps. Coupling Compendium to Brahms entails the creation of a Java class that serves, within Compendium, as a host for Brahms. Compendium's code was modified to accommodate these additional capabilities to support this framework: (i) core Compendium was modified to create the agent host class which, on booting establishes an instance of a Brahms agent ready to communicate' (ii) Compendium's user interface was extended (see below); (iii) an SQL query was added responding to requests from other agents.

A Brahms agent was programmed to operate in the IBIS environment, and an adapter framework was created that couples a Brahms instance to a Compendium. Brahms agents communicate with each other, and with their Compendium hosts. To generalize this picture, any compatible IBIS platform could be substituted for the Compendium platforms. To define terms that will be used in the following section, we refer to one Compendium-Agent pair as *local*, and other pairs as *remote*. In our user scenario, a local pair serves "Susan" as her argument-mapping platform. One or more remote pairs behave as if they are database servers capable of responding to simple queries.

Figure 8 illustrates the additional feature that provides a menu item for requesting an agent to search for nodes to satisfy a given query. Figure 9 illustrates the query being formed. Note that the '%' character serves as a wildcard: the query will match IBIS nodes that offer 'climate' inside any sentence. The query is broadcast from the local Brahms agent to available remote agents. Our first iteration restricts that broadcast to one remote agent coupled to a second instance of Compendium, the database containing other IBIS conversations. Our prototype implements a simple SQL query to the remote Compendium's database. Any node that has a statement that includes the word "climate" will be returned, as illustrated in Figure 10.

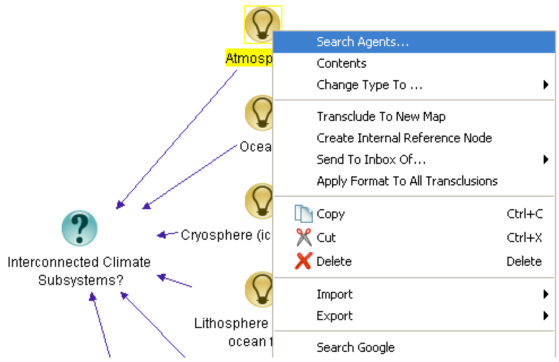

Figure 8: Compendium 'Search Agents' menu selection

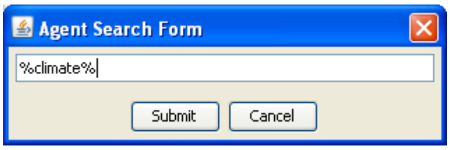

Figure 9: Composing a query

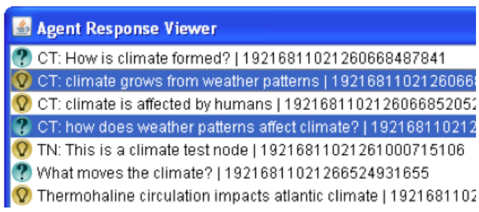

Figure 10: Search results returned from the agent network

The user can select one or more responses from the Agent Response Viewer, which leads to the addition of nodes to the original target node.

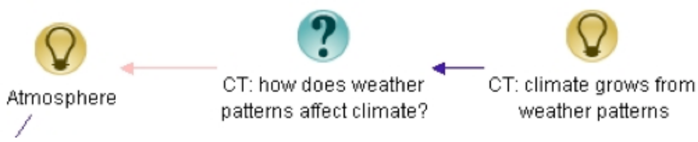

Figure 11: The local IBIS conversation expanded by query results

This serves as a proof of concept to demonstrate the feasibility of agent-mediated search within Brahms, invoked from within the Compendium client user interface, with results from a remote IBIS database being inserted, at the analyst's discretion, into her local IBIS map. We consider future developments next.

6. Conclusions and Future Work

Our research is designed towards realizing practical CSCA tools to help people cope with the complexity of knowledge intensive work in organizations, and with the challenges now facing society at large. Part of this future includes, we propose, software agents to help scaffold human reasoning: while agents can already be delegated simple tasks in order to release human effort for higher order reasoning, research in argumentation and MAS now points to the potential of human-agent argumentation.

In this paper we have motivated two investigations into how a particular agent-oriented language and architecture can augment CSCA: (i) the use of the IBIS formalism enabling Brahms agents to interact about competing options using argumentation, and (ii) the extension of the Compendium CSCA tool by integrating it with Brahms agents tasked with detecting related discourse elsewhere.

These results provide proof of the concept of human-agent argumentation. There is significant work left to develop this infrastructure to the point where we can begin to evaluate it in authentic contexts, such as the NASA field trials reviewed at the start. Brahms agents could be extended from the current simulation of Issue Mapping in IBIS, to identify when a particular kind of argument is being made, in order to deploy the IBIS argumentation schemes we have as Compendium templates [28].

Brahms already provides the low-level protocols for distributed internet-based MAS. IBIS-agents seeking to connect and federate IBIS conversations over the Internet will need to infer potential connectedness based on different (non-exclusive) strategies, currently under review [22]. One strategy seeks to exploit the contextual cues provided by IBIS structure, for instance, if two Issues in two different maps are similar, then arguably, they establish similar contexts for comparing the nodes connected to them. Another strategy would add more sophisticated language processing to parse node labels. A third strategy is to use topic mapping federation techniques.

Finally and most ambitiously, the Brahms language has been shown in a range of NASA mission contexts to be capable of modelling and implementing multi-agent systems to scaffold authentic work practices. This expands our vision for computational modelling of argumentation: it could be possible to move beyond modelling the micro-worlds of agent-agent and human-agent argumentation, to conceiving and modelling the broader work system in which this takes place, considering the actors and contexts that cause new issues to be raised, constraining the options and tradeoffs, and with an understanding of who or what may then make use of the results of argumentation.

Acknowledgements: We gratefully acknowledge the support of NASA Ames Research Center in conducting the IBIS Brahms agent research. Compendium is currently funded by the Hewlett Foundation (OpenLearn & OLnet Projects), and previously by the UK AHRC, EPSRC, ESRC and JISC.

7. References

1 Kirschner, P. Buckingham Shum, S. and Carr, C. (Eds.) *Visualizing Argumentation.* Springer: London.

2 Okada, A., Buckingham Shum, S. and Sherborne, T. (2008). *Knowledge Cartography.* Springer: London

3 Reed, C., Walton, D. & Macagno, F. (2007). Argument diagramming in logic, law and artificial intelligence. *Knowledge Engineering Review*, 22 (1), 87-109

4 Buckingham Shum, S., et al. (2006). Hypermedia Support for Argumentation-Based Rationale: 15 Years on from gIBIS and QOC. In: *Rationale Management in Software Engineering.* Dutoit, R. et al.. Springer: Berlin

5 de Moor, A. and Aakhus, M. (2006). Argumentation Support: From Technologies to Tools. *Communications of the ACM*, 49, (3), pp.93-98

6 Selvin, A., Buckingham Shum, S.J. & Aakhus, M. (In Press). The Practice Level in Participatory Design Rationale: Studying Practitioner Moves and Choices. *Human Technology Journal*

7 Sierhuis, M., W. J. Clancey, et al. (2009). Brahms: An Agent-Oriented Language for Work Practice Simulation and Multi-Agent Systems Development. *Multi-Agent Programming*, Rafael, H. et al. Springer

8 Rittel, H.W.J., Second Generation Design Methods. *Interview in: Design Methods Group 5th Anniversary Report: DMG Occasional Paper*, 1972. 1: p. 5-10. Reprinted in: *Developments in Design Methodology*, N. Cross (Ed.), 1984, pp. 317-327, J. Wiley & Sons: Chichester

9 Buckingham Shum, S. and N. Hammond (1994). "Argumentation-Based Design Rationale: What Use at What Cost?" *International Journal of Human-Computer Studies,* 40(4): 603-652

10 Rittel, H.W.J., Second Generation Design Methods. *Interview in: Design Methods Group 5th Anniversary Report: DMG Occasional Paper*, 1972. 1: p. 5-10. Reprinted in: *Developments in Design Methodology*, N. Cross (Ed.), 1984, pp. 317-327, J. Wiley & Sons: Chichester

11 Buckingham Shum, S. and Okada, A. (2008). Knowledge Cartography for Controversies. In: *Knowledge Cartography,* (Eds.) Okada, A., Buckingham Shum, S. and Sherborne, T. Springer: London

12 Conklin, J. and M.L. Begeman (1988). gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, 4 (6): p. 303-331.

13 Conklin, J. (2006) *Dialogue Mapping*. Wiley & Sons: Chichester

14 Sierhuis, M. and Selvin, A. (1996). Towards a framework for collaborative modeling and simulation. *Workshop on Strategies for Collaborative Modeling and Simulation*, ACM CSCW '96, Boston, MA

15 Selvin A. (1999) Supporting Collaborative Analysis and Design with Hypertext Functionality. *Journal of Digital Information*, 1 (4): http://journals.tdl.org/jodi/article/view/jodi-17/15

16 Sierhuis, M. and Buckingham Shum, S. (2008). Human-Agent Knowledge Cartography for e-Science. In *Knowledge Cartography*. (Eds.) Okada, A., Buckingham Shum, S. and Sherborne, T. Springer: London

17 Compendium Institute: http://compendium.open.ac.uk/institute

18 Debategraph: www.debategraph.org

19 Klein, M. and Iandoli, L. (2008). Supporting Collaborative Deliberation Using a Large-Scale Argumentation System: The MIT Collaboratorium. *Conference on Online Deliberation*.

20 Buckingham Shum, S. (2008). Cohere: Towards Web 2.0 Argumentation. *2nd International Conference on Computational Models of Argument*, 28-30 May 2008, Toulouse. IOS Press: Amsterdam

21 IBIS Interchange Specification: http://projects.kmi.open.ac.uk/hyperdiscourse/docs/IBIS-0.1.pdf

22 Park, J. (2010). Boundary Infrastructures for IBIS Federation: Design Rationale, Implementation, and Evaluation. *Technical Report KMI-10-01*, Knowledge Media Institute, Open University, UK

23 Sierhuis, M. (2007). “It’s not just goals all the way down” – “It’s activities all the way down.” *Engineering Societies in the Agents World VII*, 7th International, Workshop, ESAW 2006, Dublin, 2006, Revised Selected and Invited Papers. G. M. P. O'Hare, A. Ricci, M. J. O'Grady and O. Dikenelli. Dublin, Ireland, Springer. LNCS 4457/2007: 1-24

24 Sierhuis, M., Clancey, W.J. et al. (2009). NASA’s OCA Mirroring System: An application of multiagent systems in Mission Control. *Autonomous Agents and Multi Agent Conference* (Industry Track). Budapest

25 Clancey, W.J., et al. (2005). Automating CapCom Using Mobile Agents and Robotic Assistants. *American Institute of Aeronautics and Astronautics 1st Space Exploration Conference*, Orlando, FL

26 Searle, J., R. (1975). A taxonomy of illocutionary acts. *Language, Mind, and Knowledge*. K. Gunderson. Minneapolis, University of Minnesota. 1-29: 344-369

27 FIPA. (2002). "FIPA Communicative Act Library Specification." Retrieved 12/10, 2004, from http://www.fipa.org/repository/aclspecs.html

28 Argumentation Schemes in IBIS: http://compendium.open.ac.uk/compendium-arg-schemes.html