Ontology-based Communication Architecture Within a Distributed Case-Based Retrieval System for Architectural Designs

Viktor Ayzenshtadt[1,3], Ada Mikyas[1], Klaus-Dieter Althoff[1,3], Saqib Bukhari[3], Andreas Dengel[2,3]

[1]University of Hildesheim, Institute of Computer Science
Samelsonplatz 1, 31141 Hildesheim, Germany

[2]Kaiserslautern University
P.O. Box 3049, 67663 Kaiserslautern, Germany

[3]German Research Center for Artificial Intelligence
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
{firstname.lastname}@dfki.de

Abstract The communication and cooperation of agents is one of the key features of the multi-agent systems theory. In this work we discuss how the agents can communicate by means of applying a domain-specific ontology for the purpose of case-based retrieval of similar architectural designs. The domain ontology and the corresponding communication patterns are parts of the communication architecture of the distributed case-based retrieval system MetisCBR. We also present a vision of the results explanation component that enhances the existing architecture with own patterns and concepts and is able to recognize the corresponding contexts in search results returned by the system.

Keywords: case-based design, multi-agent systems, ontology, communication

1 Introduction

In multi-agent systems, the communicative interconnection of agents available in the system is established by providing a communication module that is able to transport messages from one agent to another. Modern FIPA-compilant[1] multi-agent frameworks like JADE [5] support the ontology-based communication of agents [7]. This allows for a convenient way of implementing a communication and cooperation component that is based on a domain-specific ontology where concepts and relations can be appropriately selected for the given task.

In this work we present the communication architecture of MetisCBR [3], the distributed case-based retrieval system for search of architectural designs,

[1] The Foundation for Intelligent Physical Agents, http://fipa.org

developed in context of the Metis project (*Metis – Knowledge-based search and query methods for the development of semantic information models for use in early design phases*).[2] This interdisciplinary project was initiated by the DFKI (German Research Center for Artificial Intelligence) and the TUM (Technical University of Munich) and unites the research areas of computer-aided architectural design (CAAD), case-based reasoning (CBR), and multi-agent systems (MAS). The project is funded by the German Research Foundation *(Deutsche Forschungsgemeinschaft, DFG).*

This paper is structured as follows: first we present the related work in the area of ontology-based agent communication. In the next section we describe the current communication architecture of the system that consists of the communication ontology and the corresponding communication patterns. After that we present a vision of results explanation module and the corresponding modification of the communication architecture. Discussion and conclusion close this paper.

2 Related Work

To date much important work has been done in the domain of ontology-based multi-agent communication. In this section we shortly describe some of the papers that we consider inspirational and helpful for conceptualization and development of our communication architecture.

The work of Steels [13] discusses the creation mechanisms of ontologies in multi-agent systems by using a number of conventions adapted from biology (self-organisation, selectionism, and co-evolution). These mechanisms are applied to the agents domain in this paper. Steels' general conclusion for *co-evolution* is especially important for the purposes of our approach: a shared ontology emerges during the communication and possesses abilities of dynamism and incompleteness – dynamism allows for the extension of the ontology by new concepts, incompleteness implies the possibility of communication with different, yet compatible definitions.

Brena und Ceballos propose in [6] a hybrid approach that combines the centralization and distribution of ontologies within a multi-agent system. In this approach, a special ontology agent plays a role of a carrier of the complete ontology and delivers the needed parts of it to other agents of the system that implement only basic parts of the ontology and make requests for needed parts when required. This approach gave us an inspiration to keep the main (meta) information of the ontology centralized, and to distribute the parts for communication and explanation into two separate ontology modules (see Section 4).

For the structure of the communication ontology (see Section 3.1), we took the work of Zhan [14] as source of inspiration. In this work the *layered* ontology is applied to the product design and analysis domain, the advantages of such a structure (extensibility and domain-oriented efficiency) are described in [14] as well. We adapted the main idea of this approach for the purposes of our domain.

[2] *Metis – Wissensbasierte Such- und Abfragemethoden für die Erschließung von Informationen in semantischen Modellen für die Recherche in frühen Entwurfsphasen.*

3 MetisCBR Agents Communication

MetisCBR is a case-based search engine for retrieval of architectural building designs that uses the application of *Semantic Fingerprint* [9] patterns to the design instances during the search. The retrieval with fingerprint patterns (fingerprints) is related to the concept of similarity footprints described in [11], the fingerprints themselves are structured by means of applying the AGraphML specification [8] to the designs. The cases (semantically transformed building designs) in the case bases of MetisCBR are built with the specific domain model described in [2].

Being a distributed system, MetisCBR contains a number of (case-based) agents that are able to communicate with each other in order to coordinate their tasks, as well as to cooperate in order to achieve their common goal (find the most similar cases for a given design query). To establish the communication between the agents and to standardize the cooperation processes inside the system, a special *communication architecture* was developed that governs the normalization of the communication process. It currently consists of the specific *communication ontology* created for the domain of retrieval of architectural building designs, and the corresponding specific *communication patterns* that are based on the retrieval tasks of the system and its agents. In the following sections we present the structure of the communication ontology and the structure of the basic retrieval communication pattern.

3.1 The General Structure of the Communication Ontology

The communication ontology is based on the concepts of MetisCBR's domain model, but contains some additional features (for example, a number of concepts that are specific only for some particular agents or agent groups). The communication ontology is divided in three different layers (see Figure 1), where each of them is used during the corresponding step of the retrieval process:

– *Object Layer* – This layer represents the general concepts of the *query* and *result* objects that are being *received from* or *sent to* the user as the object that is created or parsed by the user interface that is connected to MetisCBR. Thus, this layer is used in the first and the last step of the retrieval.
– *Data Layer* – In this layer the query and result objects are decomposed into the data representations according to the CBR domain model [2] of the retrieval system. The architectural design concepts FLOORPLAN, ROOM and EDGE from the model will be represented with their corresponding ontological equivalents *(metadata, rooms and edges data)* and will be used during the actual retrieval steps of the complete retrieval process.
– *Action Layer* – This layer is responsible for representation of categories of actions that agents of the system are able to execute. For example, to parse and transform the query object into an ontological representation, resolve the query using the given retrieval strategy, forward the query (or its parts) to other agents, or to construct and save a concept instance that will be used to represent a case for the retention component of the coordination agent.

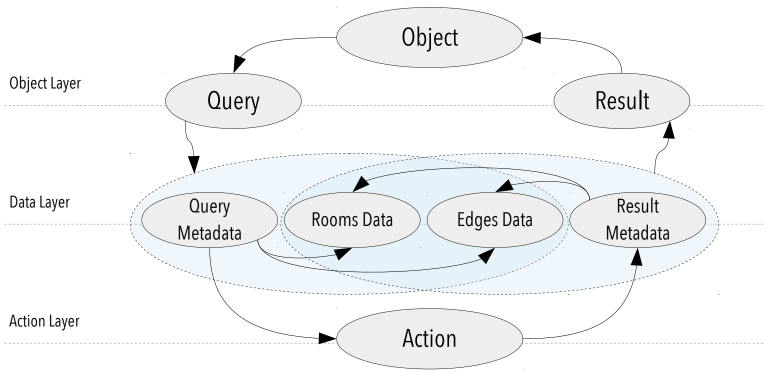

Figure 1. The current general structure of the communication ontology.

To utilize the ontology in order to communicate with each other, the agents of the system use communication patterns that are based on the concepts of the ontology. Communication patterns consist of steps that are named after the *action class* that contains the action the agent is requested to execute. The patterns can contain further sub-patterns. Following components are required for construction of a communication (sub-)pattern:

– *Action Class* – The category of the action to be assigned. Strategical system restrictions specify which actions an agent is free to execute when requested.
– *Actor* – The local identification address of the agent that is requested to perform the selected action.
– *Purpose* – The goal(s) of the action the agent is requested to accomplish, if it has committed or was assigned to this task.
– *Content* – An ontological object (for example the rooms data or a list of result floorplan IDs) that the agent uses as information source to accomplish its task. Can also contain further objects or references to objects.

3.2 Retrieval Communication Pattern

In this section we demonstrate how the agents communicate with each other using the communication patterns of the system. We show it by providing a detailed description of the steps of the general *retrieval communication pattern*. The communication flow of this pattern involves almost every agent type available in the system (except the case base maintainer agent). This pattern is the basic pattern of the communication and cooperation and uses almost all of the available action classes and ontology concepts to establish the undisturbed communication process during the retrieval. Figure 2 shows the graph-based representation of the structure of the pattern that consists of the following steps:

– *XHR* – The purpose of this action class is the transmission of the user query in XML format for later parsing and resolving. The actor (receiver of the query) in this case is the coordination agent (denoted as *Coord.* in Figure 2).

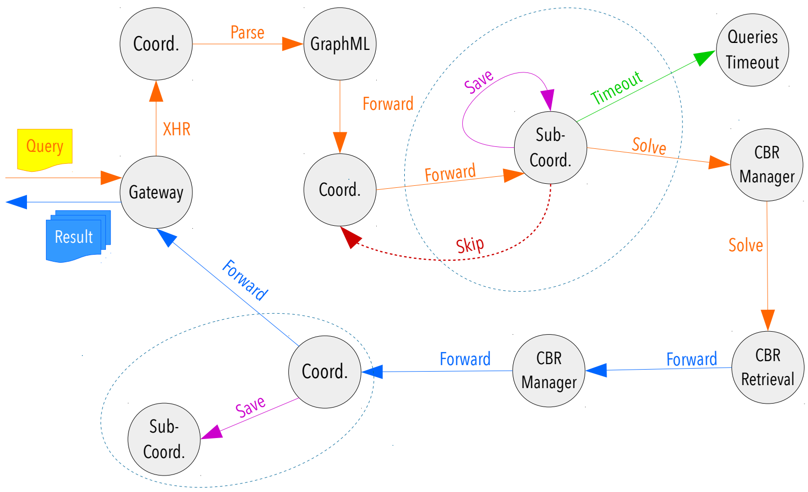

Figure 2. The graph-based representation of the *retrieval communication pattern*. The node labels denote the agents, the edge labels denote the action classes (steps).

- *Parse* – The receiver (actor) of the parsing request is the corresponding parsing agent (GraphML) that transforms the XML-formatted query into the ontological query in *SL* language format.
- *Forward* – This action class is intended to be used for forwarding of the content (ontological query or result) to other agents. Any agent can be a sender or receiver of this kind of task.
- *Solve* – The purpose of this action category is to request an appropriate agent to resolve the ontological building design query (or its parts) and to find results in a case base of such designs. A manager agent or a retrieval agent can be requested to accomplish this task.
- *Timeout* – The *Timeout* agent that receives the request containing this action class is asked to add the current retrieval task to the list of *active* tasks and to *check periodically* if the query is expired (not resolved during the given amount of time).
- *Skip* and *Save* – These two classes are related to each other and are intended to be used in the *case learning* sub-pattern (enclosed in the dashed areas in Figure 2), which is a sub-pattern of the *retrieval communication pattern* and is used during the retention step of the Coordinator's internal CBR cycle, which is the step that helps to find the most similar previous query to the current one (this task is delegated to the *SubCoordinator*). The internal case base of Coordinator consists of the previous queries and is filled by means of applying the IB2 algorithm [1], in the way that only unique queries (and corresponding results) will be saved there. If the current query is identical to a previous one, the *Skip* action will be sent to the Coordinator to indicate this fact. The *Save* action class will only be used if the current query is *not identical* to a previous one. In this case the query will be saved before the actual retrieval starts, the results after the retrieval has been finished.

4 Vision of Results Explanation Module

Explanations are one of the core elements in user-centered CBR applications. Foundations, perspectives, and goals of explanations in CBR are described in [10] and [12]. In this section of the paper we present our vision of the extension of the MetisCBR system with an explanation module (see Figure 3) that contains its own explanation ontology. This module is currently being conceptualized in a bachelor thesis. Our general idea is to *combine two separate ontology modules* (the communication ontology and the new *explanation ontology*) into a *system ontology* (where the main meta information about these modules is kept permanently), but to use their concepts separately for the corresponding communication and explanation tasks. The explanation ontology will be used for the corresponding *explanation patterns* (that will have a structure similar to the communication patterns described in Section 3.2) and connected to a specific *explanation engine* that can use this patterns to work with different contexts (i.e., recognize if some results have one or more contexts as common criteria) and return an explanation of the retrieval results (based on these recognized contexts) to the user (an architect). The contexts can represent different semantic fingerprints or other criteria (for example, some of the floor plan results can belong to the same building, i.e., have the common building ID). It should be possible to have the permanent contexts (saved in the explanation ontology) as well as the temporary contexts that are specific only for the current search process.

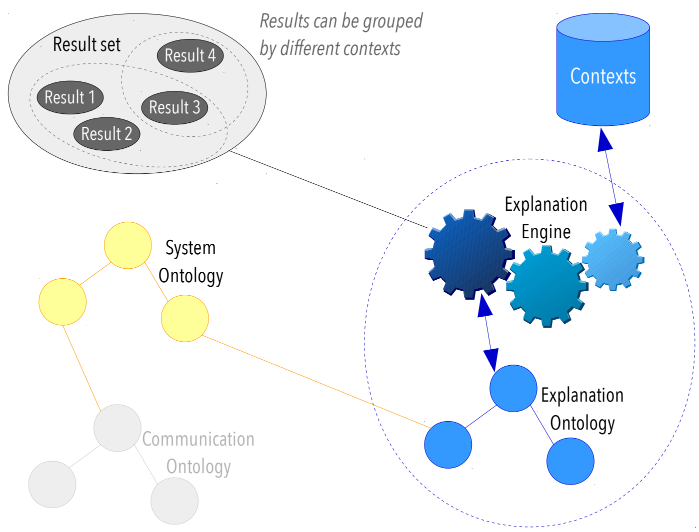

Figure 3. The current vision of the results explanation component for MetisCBR.

5 Discussion

The whole potential of the ontology-based agent communication architectures is not fully explored, but is used often to provide a basis for template- or pattern-based communication and cooperation among the agents contained in the system. In our retrieval system the ontology plays a role of the *layer-structured* relational vocabulary of objects and corresponding action classes that can be used by appropriate agents to request an action that needs to be executed for the current retrieval task.

In the evaluations of MetisCBR conducted to date (for example in [2] and [4]), and also during the development process, the ontology-based communication architecture showed a good performance (currently the size of the communication ontology does not allow for conducting of the performance test for the ontology only, so that the performance could only be estimated in context of the complete retrieval process, but no technical ontology-related issues worthy of mention were detected during the evaluations). The clearest advantage of such an architecture is the possibility to extend and restructure the underlying structure of concepts and actions by adding the new ones and/or deleting/editing the currently available ones. Though extensible, a certain technical limitation of the ontology scope exists as well, characterized by non-extensibility of the number of actions available for each of the agents at the runtime of the system.

6 Conclusion and Future Work

In this paper we presented the current communication architecture of MetisCBR, a distributed case-based retrieval system for search of semantically represented architectural designs. In this architecture the communication ontology plays a key role by providing a communication and cooperation basis for the agents of the system. We showed that the communication relies on the special communication patterns and provided and explained in detail an example of such a pattern (the basic *retrieval communication pattern*). We also provided our idea of how the concept of such a communication architecture can be used and adapted by an explanation module that is able to detect certain contexts in the result sets returned by the retrieval system, and how we can combine the ontologies of both communication and explanation.

In our future work on our case-based retrieval system we will concentrate on finalizing of the conceptualization of the above named explanation component and include it as a permanent part of the retrieval system and the corresponding retrieval process. Elaboration and extension of the available contexts, in order to improve the context recognition, will also be part of our future work in this area. The further development of other parts of the retrieval system, for example, implementation of new retrieval methods, will also be continued.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine learning 6(1), 37–66 (1991)
2. Ayzenshtadt, V., Langenhan, C., Bukhari, S.S., Althoff, K.D., Petzold, F., Dengel, A.: Distributed domain model for the case-based retrieval of architectural building designs. In: Petridis, M., Roth-Berghofer, T., Wiratunga, N. (eds.) Proceedings of the 20th UK Workshop on Case-Based Reasoning. UK Workshop on Case-Based Reasoning (UKCBR-2015), located at SGAI International Conference on Artificial Intelligence, December 15-17, Cambridge, United Kingdom. School of Computing, Engineering and Mathematics, University of Brighton, UK (2015)
3. Ayzenshtadt, V., Langenhan, C., Bukhari, S.S., Althoff, K.D., Petzold, F., Dengel, A.: Thinking with containers: A multi-agent retrieval approach for the case-based semantic search of architectural designs. In: Filipe, J., van den Herik, J. (eds.) Proceedings of the 8th International Conference on Agents and Artificial Intelligence. International Conference on Agents and Artificial Intelligence (ICAART-2016), February 24-26, Rome, Italy. SCITEPRESS (2016)
4. Ayzenshtadt, V., Langenhan, C., Roth, J., Bukhari, S.S., Althoff, K.D., Petzold, F., Dengel, A.: Comparative evaluation of rule-based and case-based retrieval coordination for search of architectural building designs. In: Goel, A., Roth-Berghofer, T., Diaz-Agudo, B. (eds.) Case-based Reasoning in Research and Development. International Conference on Case-Based Reasoning (ICCBR-16), 24th International Conference on Case Based Reasoning, October 31 - November 2, Atlanta„ Georgia, USA. Springer, Berlin, Heidelberg (2016)
5. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE, vol. 7. John Wiley & Sons (2007)
6. Brena, R., Ceballos, H.: A hybrid local-global approach for handling ontologies in a multiagent system. In: Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference. vol. 1, pp. 261–266. IEEE (2004)
7. Caire, G., Cabanillas, D.: Jade tutorial: application-defined content languages and ontologies. TILab SpA (2002)
8. Langenhan, C.: A federated information system for the support of topological bim-based approaches. Forum Bauinformatik Aachen (2015)
9. Langenhan, C., Petzold, F.: The fingerprint of architecture-sketch-based design methods for researching building layouts through the semantic fingerprinting of floor plans. International electronic scientific-educational journal: Architecture and Modern Information Technologies 4, 13 (2010)
10. Roth-Berghofer, T.R.: Explanations and case-based reasoning: Foundational issues. In: European Conference on Case-Based Reasoning. pp. 389–403. Springer (2004)
11. Smyth, B., McKenna, E.: Footprint-based retrieval. In: Case-Based Reasoning Research and Development, pp. 343–357. Springer (1999)
12. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in case-based reasoning–perspectives and goals. Artificial Intelligence Review 24(2), 109–143 (2005)
13. Steels, L.: The origins of ontologies and communication conventions in multi-agent systems. Autonomous Agents and Multi-Agent Systems 1(2), 169–194 (1998)
14. Zhan, P.: An ontology-based approach for semantic level information exchange and integration in applications for product lifecycle management. Ph.D. thesis, Citeseer (2007)