A Broker Architecture for Integration of Heterogeneous Applications for Inquiry Learning

Lars Bollen[(1)], Andreas Harrer[(1)], H. Ulrich Hoppe[(1)] and Wouter van Joolingen[(2)]
[(1)]*University Duisburg-Essen, Germany*
{bollen, harrer, hoppe}@collide.info
[(2)]*University of Twente, The Netherlands*
W.R.vanJoolingen@utwente.nl

Abstract

In the context of the CIEL activity within the European network of excellence Kaleidoscope, a generic software architecture has been developed to integrate various heterogeneous applications for inquiry learning, including the definition of relevant data formats and interfaces. The general aim of this architecture is to provide a common software platform for various learning applications, facilitating interoperability by means of quasi-synchronous data exchange. Dynamically created learning objects comply with the LOM standard, thus enabling the re-use of these product in follow-up learning activities.

1. Introduction

Scientific Discovery Learning [1] and Inquiry Learning [2] both aim at letting the learners explore problems from a researcher's perspective. This kind of learning changes also the style of support that computer tools can provide: Delivery of learning material becomes less relevant, while interactive support for the learners becomes more important. Discovery and inquiry processes usually consist of the design of an experiment, the execution, and evaluation of the experiment and – depending on the evaluation – a repetition of this sequence, i.e. a cyclic process. In a more fine-grained perspective [3] the generation of hypothesis and predictions are also mentioned as important steps. In computer-supported learning environments experiments can be conducted with the help of computer simulations [4], thus avoiding high costs for conducting real experiments with expensive material or making feasible experiments that would be dangerous to conduct directly (such as with radioactivity or toxic substances).

Kaleidoscope's CIEL (Collaborative Inquiry and Experiential Learning) activity [5] targets both the conceptual as well as the technical integration of different perspectives on the inquiry learning process and its phases. The technical perspective aims at capitalizing on the different existing inquiry learning support tools, each of which has specific application possibilities (e.g., in terms of phases or learning scenarios) By making these tools interoperable with each other the support of the whole inquiry process will be facilitated by using the best suited tools for each activity in combination through the whole process.

2. Approach

The general aim of the CIEL architecture is to provide a common software platform for a heterogeneous set of inquiry learning tools with partly overlapping and partly complementary functionalities. The goal is interoperability in terms of the exchange and further processing of data sets or models generated by one of the tools. The basic cooperation mode would be asynchronous because it gives higher flexibility, but short update intervals should also allow for quasi synchronous exchange. To be open for easily incorporating new tools and extending the scope of the integration environment, we favor a loosely coupled architecture for data exchange with a central broker supporting standardized data formats.

The following list enumerates the basic ingredients of the CIEL approach:

- a conceptual data model of relevant entities based on a task ontology (including "experiments", "data sets", "models" etc.),
- user and profile management functions (including grouping of users and the specification of interest profiles),
- a subscription/notification mechanisms for data transfer based on interest profiles.

Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)
0-7695-2916-X/07 $25.00 © 2007 IEEE

In the next sections, we will describe the implementation based on a broker architecture and well-defined data formats for coupling different learning applications, independent of programming languages, operating systems or locations.

3. Broker Architecture

In general, a broker is responsible for the coordination between participants in a network [6]. Considering the premises described above, a broker architecture is ideally suited to meet the specific requirements because it avoids a one-to-one interfacing of the different tools. When various learning environments (LE) are used in an inquiry process, a broker architecture is able to mediate between these LEs and to handle data exchange.

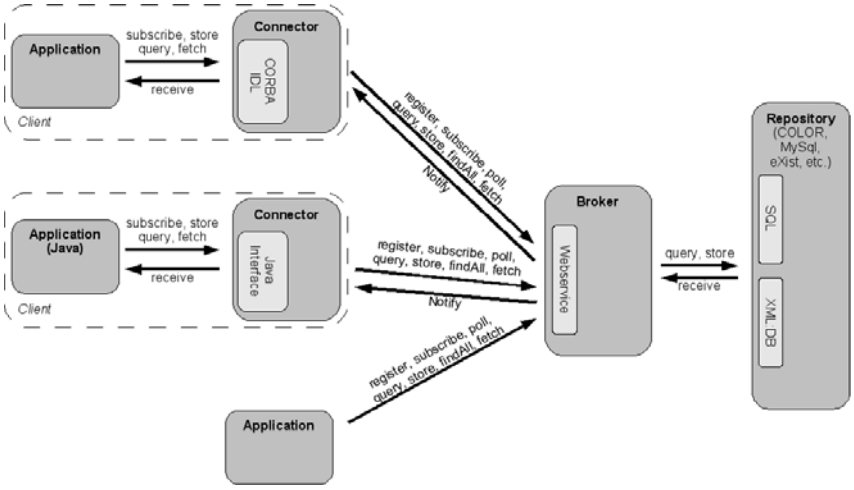

Figure 1: Overview of the Broker architecture

After registering to the broker, a client can store documents by sending them to the broker. Clients can query for documents or subscribe to specific types of content (on arrival/update). Normally, communication between the broker and an LE is handled through a connection module that is provided in our architecture. Nevertheless, since all communication is based on web services, it is also possibility to access the broker directly in case a connection module is unwanted or not sufficient. The application then has to handle all work normally done by the connectors.

Every time the broker receives a new data item, it is stored into a repository. At the moment, SQL or XML databases are supported. The module (broker connection module) provides a thin interface for the learning environments. There are two different ways to access these modules. In addition to a Java interface, we also provide a solution in CORBA [7]. The main feature of CORBA is to provide interoperability across heterogeneous implementation languages and operating systems. The Java connector was defined for use with the existing applications Co-Lab, FreeStyler and another Java-based inquiry support tool. Figure 1 shows an overview of the architecture's components.

For transparency reasons, the connector modules handle the registration at the broker and the transfer of data items, i.e. every application can use the broker without knowledge about the broker's location or about any other client application. Finally the repository storing the data items can be inquired by an application directly to make the results of a broker network available for further use.

4. Transfer of learning products by use of LOM

Learning activities in the different LEs are "productive" in the sense that they generate new learning objects. We call these "emerging learning objects" to distinguish them from predefined materials [8]. The description of learning objects uses the conceptual data model for inquiry learning. For CIEL, we have defined several data types for learning products, such as data sets and graph models. Additionally, we support the more formal parameters of the IEEE LOM (Learning Object Metadata) standard [9] to be compatible with other platforms and applications. For the indexing of the created learning objects the broker automatically fills a set of LOM properties, mainly associated with technical aspects, while the more content-oriented and pedagogical properties (see

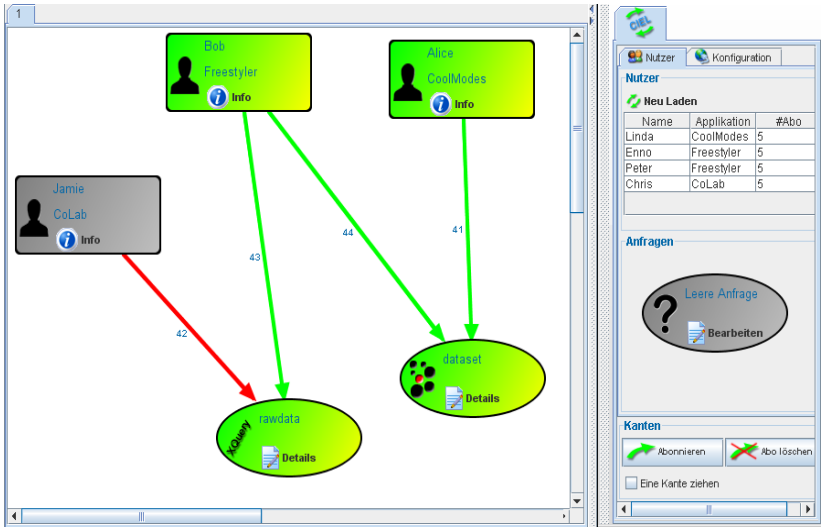

Figure 2: Visual broker administration

Table 1) are provided by the inquiry tools. Thus only a very small set of LOM attributes has to be provided manually by the learners or by an expert taking care of repository contents. This follows from the experience that explicit indexing of LOs by the users usually fails due to the high effort of creating metadata [10].

Table 1: Fragment of CIEL's data model with LOM

```
<ciel><cieldata>
<ims:lom>
<ims:technical><ims:format>ciel/dataset</ims:format>
  <ims:requirement><ims:name>
     <ims:source>ciel:software</ims:source>
     <ims:value>Freestyler</ims:value>
   </ims:name></ims:requirement>
</ims:technical>
</ims:lom>
<data>
  <dataset>
    <value>Arbre-8-96</value>
    <value>313.504544089716</value>
  </dataset>
</data>
</cieldata></ciel>
```

5. Queries and Subscriptions

The broker is able to handle queries for documents as well as subscriptions. A query for a document consists basically of a document that has no content but has relevant LOM sections (see section above) filled in. When querying the broker, the LOM attributes of the query are matched with the repository content, and a list of document URIs is returned. E.g. a query for a dataset to be used with the FreeStyler application would result in a list containing the learning object in Table 1.

Subscribing for certain learning objects work similar: An empty document with relevant metadata is given to the broker. In return, the broker notifies subscribed clients about new documents.

Along with the broker, a Freestyler plugin has been developed to facilitate visual broker administration (see Figure 2). With this plugin, a teacher is able to create queries and subscriptions and bind them to registered users. E.g., the students Alice & Bob are subscribed to datasets produced by their classmates, because they have been assigned the roles of analysts in an inquiry team.

Therefore, a teacher is in full control over the data flow in a classroom situation. These configurations can be stored and retrieved for later usage.

6. Perspectives

The data flows a teacher specifies with the visual interface implicitly also define a process sequence. If these aspects should become more explicit and the learning scenario should be scaffolded by hints, materials, and additional tools, it is feasible to combine the data-flow oriented architecture with formal learning process models or scripts. These have been discussed in the last years under the notion of Educational Modelling Languages (EMLs). We used automated mappings of graphical notations to the most prominent EML, the IMS Learning Design (IMS/LD) language [11], so the transfer and integration of our graphical broker administration with EMLs is a natural choice for future work.

7. References

[1] de Jong, T., and van Joolingen, W. (1998). "Scientific Discovery Learning with Computer Simulations of Conceptual Domains." *Review of Educational Research*, Vol. 68, No. 2, pp. 179-201.

[2] Linn, M. C., and Slotta, J. D. (2000). "The Knowledge Integration Environment: Helping students use theInternet effectively" In: Learning the Sciences of the 21st Century, Lawrence Erlbaum Associates, pp. 193-226.

[3] Friedler, Y., Nachmias, R., and Linn, M. C. (1990). "Learning scientific reasoning skills in microcomputer-based laboratories." *Journal of Research in Science Teaching,* 27(2): 173–191.

[4] Veermans, K., van Joolingen, W., an de Jong, T. (2006). "Use of Heuristics to Facilitate Scientific Discovery Learning in a Simulation Learning Environment in a Physics Domain." *International Journal of Science Education,* Vol. 28, No. 4 / 18, pp. 341-361.

[5] http://www.cielproject.eu, last visited Feb., 23rd, 2007.

[6] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad , P. and Stal, M. (1996). "A System of Patterns", John Wiley & Sons, Chichester.

[7] Siegel, J. (1996). "CORBA Fundamentals and Programming", Wiley & Sons, New York.

[8] Hoppe, H.U.; Pinkwart, N.; Oelinger, M.; Zeini, S.; Verdejo, F.; Barros, B.; Mayorga, J.I. (2005). "Building Bridges within Learning Communities through Ontologies and 'Thematic Objects'". In: Proc. of CSCL2005, Taiwan.

[9] Learning Object Metadata Standard, IEEE. http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html, last visited Feb., 28th, 2007.

[10] Cardinaels, K., Meire, M., Duval, E. (2005). "Automating metadata generation: the simple indexing interface." In: Proc. of WWW2005, pp. 548-556.

[11] Harrer, A., Kobbe, L., and Malzahn, N. (2007). "Conceptual and Computational Issues in the Formalization of Collaboration Scripts". Accepted for Proc. of CSCL07.

Acknowledgements

The following people have considerably contributed to this work: B. Hassing, J. Crespo (Univ. Duisburg-Essen), A. Krzywinksi (Univ. of Bergen), J. Toussaint (Univ. of Oslo), J. Sikken (Univ. of Twente).