Scheuer, O., McLaren, B.M., Loll, F., & Pinkwart, N. (2009). An Analysis and Feedback Infrastructure for Argumentation Learning Systems. In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED-09)*, Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling. (pp. 629-631). IOS Press.

An Analysis and Feedback Infrastructure for Argumentation Learning Systems

Oliver SCHEUER[a], Bruce M. MCLAREN[a,b], Frank LOLL[c], Niels PINKWART[c]
[a] *German Research Center for Artificial Intelligence (DFKI), Germany*
[b] *Carnegie Mellon University, Pittsburgh, PA, U.S.A.*
[c] *Clausthal University of Technology, Germany*

Abstract. In this paper, we discuss design considerations and our plans to develop a generalized framework for intelligent support in educational argumentation systems. Our goal is to develop a framework that will support the development of argumentation learning systems across a variety of domains (e.g., the law, ethics).

Introduction

The use of educational technology to support the learning of argumentation has become increasingly common in schools and universities resulting in considerable efforts in developing and assessing educational argumentation systems in recent years (e.g., [1]). Many of these efforts have been effective for specific argumentation domains but not much research has been targeted at generic, flexible, and reusable software architectures for building such tools. Being able to build upon a software architecture that provides many of the basic, generic features required for argumentation has the potential to reduce development time significantly, as compared to "from scratch" development approach. Our LASAD project[1] tackles this issue by developing a generalized framework and methodology for the construction of argumentation systems to help students learn argumentation in different domains.

But what are the essential features and requirements of a flexible software architecture for the implementation of a variety of potentially differently targeted educational argumentation systems? We have conducted an extensive literature review of existing systems and approaches, covering general-purpose tools, educationally targeted tools, as well as problem-solving environments with supplementary discussion facilities. Elsewhere we discuss overall architecture requirements [2]; here we focus on the infrastructure requirements for flexible intelligent learning support that result from our review.

1. Requirements and Design Considerations

The central objective of our approach is to provide infrastructure that can be (re-)used across multiple domains, hence **generality** is one of our central concerns. The architecture should be capable of supporting a wide variety of analysis and feedback functions. Components should detect and remedy weaknesses specific to particular argumentation domains (e.g., circular arguments in scientific argumentation [3]), problem/task instances (e.g., the non-consideration of crucial passages of a given transcript [4]) and problems in student interaction (e.g., imbalance/lack of participation, off-topic discussion [5]; failed knowledge sharing [6]). There are two basic kinds of analyses that should be possible within our framework, namely action-based analyses of log data traces [6] and state-based analyses of snapshots of arguments [5].

[1] "Learning to Argue: Generalized Support Across Domains", project funded by the German Research Foundation (DFG), project website: http://lasad.dfki.de/

Based on the indicators and diagnoses resulting from the different kinds of analyses, feedback should be provided to scaffold students' learning. Feedback can be provided in many different ways, and often it is unclear which strategy is preferable to achieve the intended learning objectives [7, 8]. Hence, the architecture should support a variety of feedback forms and strategies including: textual feedback and highlighting of graphical elements [3, 4, 5]; immediate [9], requested [3, 4] and delayed feedback [9]; feedback provided to the group and to individuals; strategies to prioritize and select feedback [3, 4].

The system should be composed of modules with self-contained functionality, clear responsibilities and interfaces. Apart from being a general characteristic of good software design, a **modular design** is especially important for our general framework because we cannot anticipate all possible application scenarios. To allow new analysis modules to be easily plugged into the framework we need to achieve **interoperability** between modules via standardized interface types and data objects. To further increase the flexibility, we aspire to a **loose coupling** between modules that can be realized via event-driven communication based on the publish-subscribe pattern, following the general approach of [10].

The run-time behavior of the intelligent support engine should be easy to set up using pre-defined "switches" of the relevant modules, realized via configuration files or by a dedicated user interface for **configuration** (as done, for instance, in [11]). Especially for the feedback strategies this is relevant: For instance, such a setting might define for a given pattern in an argument diagram (e.g., a circular argument) the feedback text to provide, whether the pattern should be visually highlighted, a priority value, etc. This information can then be used during runtime to decide whether and how to provide feedback. Thereby it will be possible to adapt the system's feedback behavior to new application scenario without programming skills, or to iteratively test and adjust different strategies without touching the source code. A further extension would be to support the configuration of controlled experiments by assigning different feedback strategies to different student groups.

2. Architecture proposal

Figure 1 sketches the most important components of the envisioned system architecture. The box on the left depicts the end user environment ("EUE"), which is used by the students to debate with one another and/or to analyze existing arguments. The specific tool is not important; it could be a chat, a threaded or a graphical argumentation system. The only requirement is that the tool complies with a general input/output interface that provides functionality to log actions to the "Data Service" and to accept feedback specifications from the outside. Components can subscribe to the "Data Service" to receive new user actions, or can request argument snapshots. The "Intelligent Support Service" comprises an "Analysis Controller", a "Feedback Controller", an arbitrary number of "Analyzer" modules, the "Feedback Selector" and the "Feedback Generator." The two controller components coordinate the analysis and feedback activities and function as event-passing hubs.

The "Analyzer" modules are not part of the framework; they can be plugged in as needed. They might analyze data locally or be local proxies for remote analysis services. The "Feedback Controller" subscribes at the "Analysis Controller" to analysis result types that are relevant for feedback provision. It also decides timing and frequency of feedback to avoid flooding students with a vast amount of messages. In case of multiple possible feedback interventions the "Feedback Selector" decides their priority so that the “Feedback Controller” can focus on the most crucial one(s). The “Feedback Generator” decides the actual feedback form (e.g., text and/or highlighting) and content (which might differ depending on whether it is the first, second, etc. time a feedback condition is fulfilled). The specification of the chosen feedback is then provided via the “Data Service” to the “EUEs” which realize the feedback in their specific ways. We intend to make the feedback modules configurable via external XML files (feedback content and strategy).

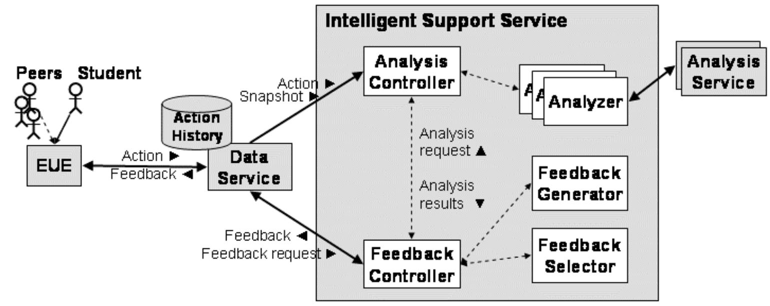

Figure 1: Proposed architecture: Solid lines indicate remote, dashed lines local communication paths.

Depending on the concrete setup multiple control flows are possible: Feedback might be explicitly requested by a student, triggered in reaction to a student action in the “EUE” (including summative feedback after closing a session) or triggered after a “routine check” of critical conditions initiated by the “Feedback Controller.”

Once the “Intelligent Support Service” is implemented, we intend to test its generality in different argumentation domains (ethics, legal argumentation, scientific argumentation) using different analysis modules (e.g., syntactical analysis [4], application of machine-learned classifiers [5]) and support strategies.

References

[1] Andriessen, J. (2006). Arguing to Learn. In: Sawyer, R.K. (Ed.), *The Cambridge Handbook of the Learning Sciences*, Cambridge University Press, New York, pp. 443-460.

[2] Loll, F., Pinkwart, N., Scheuer, O. & McLaren, B.M. (2009). An Architecture for Intelligent CSCL Argumentation Systems. In *Proc. of the 8th Int’l Conference on Computer Supported Collaborative Learning (CSCL-09).*

[3] Suthers, D. D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E. E., Toth, J., Weiner, A. (2001). Representational and Advisory Guidance for Students Learning Scientific Inquiry. In *Smart machines in education: The coming revolution in educational technology*, pp.7-35.

[4] Pinkwart, N., Ashley, K., Lynch, C., & Aleven, V. (2008). Graph Grammars: An ITS Technology for Diagram Representations. In *Proc. of the 21st Int’l FLAIRS conference*, AAAI, pp. 433-438.

[5] McLaren, B.M., Scheuer, O., De Laat, M., Hever, R., De Groot, R., & Rosé, C.P. (2007). Using Machine Learning Techniques to Analyze and Support Mediation of Student E-Discussions. In *Proc. of the 13th Int’l Conference on AI in Education (AIED 2007)*, IOS Press, (p. 141-147).

[6] Soller, A. Understanding knowledge sharing breakdowns: A meeting of the quantitative and qualitative minds. In *Journal of Computer Assisted Learning*, 2004, 20, pp. 212-223.

[7] VanLehn, K. The behavior of tutoring systems. In *Int’l Jour. of AI in Education*, 2006, 16, pp. 227-265.

[8] Shute, V. J. Focus on Formative Feedback. In *Review of Educational Research*, 2008, 78, pp. 153-189.

[9] Israel, J. & Aiken, R. Supporting Collaborative Learning With An Intelligent Web-Based System *Int’l Journal of Artificial Intelligence in Education*, 2007, 17, pp. 3-40.

[10] Pinkwart, N. (2003). A Plug-in Architecture for Graph Based Collaborative Modelling Systems. In: U. Hoppe, F. Verdejo & J. Kay (eds.) *Proc. of the 11th Int’l Conference on Artificial Intelligence in Education* (AIED 2003), (p. 535-536).

[11] Ravenscroft, A., Sagar, M., Baur, E & Oriogun. P (2008). Ambient pedagogies, meaningful learning and social software. In Hatzipanagos, S. & Warburton, S. (eds.), *Social Software & Developing Community Ontologies*, IGI Global Publishing, pp 432-450