An Integrated Approach for Analysis-Based Report Generation

Lars Bollen, Andreas Harrer, H. Ulrich Hoppe
University Duisburg-Essen
47047 Duisburg
{bollen, harrer, hoppe}@collide.info

Abstract

Today's learning and modeling environments mostly provide support for storing and presenting only results of learning and modeling processes, not the process itself. Since self-reflection, assessment, authoring and research are highly dependent on reporting this process information properly, we propose an intelligent approach for generating reports automatically based on the analysis of learning protocols.

1: Introduction: Rationale & use of reporting

In this paper we consider "reports" as summaries of action traces from learning (specifically: modeling) processes, that are interpreted, augmented and visualized in different ways and for various purposes. Such reports are useful, since usually only results can be stored for re-use and documentation; while details of learning / modeling processes (e.g. phases, milestones, design rationale) get lost. Depending on the role of the report's recipient, the deficiency mentioned above can be solved by offering specific kinds of reports:

a) Reports for learners: Is the learner himself the recipient, a report can support self-reflection (when the underlying logfiles have been created by himself), vicarious learning (when the logfiles have been created by other learners) and presenting his results.
b) Reports for teachers: When targeting teachers with a report, it can support assessment (when the data origins from learners) and authoring (here the action traces can come from learners or from the teacher himself).
c) Reports for research: Focusing on various aspects of CSCL (and in particular focusing on modeling), reports can help interpreting and understanding the learner's actions from a scientific point of view [1].

This classification contains all the recipients' roles we want to support directly, but may not cover every situation that might occur.

2: Approaches for reporting

A first approach to support reporting and documenting a modeling process is recording and replaying the whole process. This has been done in the field of remote courses by using streaming media of different types. These records include various subsets of video or audio streams, slides with additional material like screenshots, modeling tools etc. These records can be watched synchronously or asynchronously. Although this material is manually post-processed sometimes, there is very little "intelligent" support in it. The modification of the original content is (mostly) limited to cutting phases of irrelevant actions or adding helpful material. Systems using this approach are e.g. "Authoring on the Fly" [2] or "E-Chalk"[3].

More recent research goes into the direction of "vicarious learning" [4]. This goes along with watching recorded material or somebody else's learning process.

Since we are focusing on modeling environments, which allow for creating, exploring and manipulating scientific models and simulations, a streaming approach is not sufficient, because streaming media does not provide interaction of the reviewing user with the captured media.

In today's modeling environments (e.g. Stella [5], Cool Modes [6], CoLab [7]), reporting and automated authoring is supported only marginally. Some are using some kind of streaming approach (e.g. the MatchMakers replay feature [8]) or support the user in manual reporting and authoring, i.e. the user has to create reporting documents explicitly.

3: An integrated approach for analysis-based report generation

A different, yet potentially more powerful, approach integrates analysis of learners' logfiles with automated generation of reports. This approach aims at identifying the essential episodes, milestones, results, collaborative situations, ownership and influence spheres, to present them in a report fitting the interests of the specific recipients mentioned above.

To achieve this, we use logfiles (e.g. low level learners' action traces) as basic information to be analyzed and processed to higher level information useful for reporting.

3.1: Logfiles

The protocols / logfiles of learners' (or teachers') interaction with the system and with other users (via the system) provide a basis for generating reports in our

approach. Therefore it is most important to have rich information about each single action in a modeling process. We assume that logfiles contain (at least) the following information:

- creation of an object (reference to object, user's name, creation time, object-specific data)
- modification of an object (reference to object, user's name modifications done on object)
- deletion of an object (reference to object, user's name)

This information can typically be captured by most of the modeling environments of the state of the art (such as Belvedere [9], CoLab [7], Cool Modes [6], ModellingSpace [10]).

In recent work we showed how to integrate additional, domain-specific knowledge into logfiles to classify user actions into categories like "meaningful", "simulation relevant" or "learner specific" [11].

Additionally, collaboration-specific data like chat recordings, annotations or remote pointer usage could be integrated into the logfiles to help analyzing and interpreting collaborative modeling.

3.2: Analyzing

The logfiles mentioned above can be parsed and analyzed in different ways: Abstracting, filtering, querying and augmenting this data (Ref.) are methods to help identify typical student behaviors, recognize phases and episodes in learning and modeling processes, evaluate collaboration etc. as we claimed above.

For this purpose, we have implementations for each of these techniques which will be refined, extended and generalized in future work:

- For Example the user can specify patterns that are searched for in logfiles and instances of these patterns are visualized.
- Logfiles can be augmented to emphasize specific user behavior when replaying a learning session.
- A method for abstracting logfile entries from a low level format, which is hard to understand, to a high level format that describes user actions in a more abstract and informative way that may include domain-specific knowledge has been implemented prototypically.
- Future work will deal with "pattern mining" to find non-specified, but typically occurring user behavior.

3.3: Generating reports

The results of these analysis techniques have to be assembled and visualized considering the report's use case.

Basically, you can distinguish between feeding the results back into the learning / modeling environment or reporting them in an external platform. Feeding the results back into the originating system is essential for authoring, since the authored material is designed to be used in exactly this environment. Feeding the results back is useful for learners' self-reflection, to avoid media-breaks and have immediate feedback. In contrast to this, an external tool dedicated for assessment or research might be more appropriate.

4: References

[1] Zumbach, J., Muehlenbrock, M., Jansen, M., Reimann, P. Hoppe, H.U. (2002). *Multi-Dimensional Tracking in Virtual Learning Teams - An Exploratory Study*. In Stahl (ed). Proc. of Computer Supported Collaborative Learning 2002, (pp. 650-651). Lawrence Erlbaum Associates, Inc. Hillsdale, N.J., USA.

[2] Müller, R., Ottmann, T. (2000). The "Authoring on the Fly" System for Automated Recording and Replay of (Tele)presentations, Special Issue on "Multimedia Authoring and Presentation Techniques" of *ACM/Springer Multimedia Systems Journal, Vol. 8, No. 3*, May 2000.

[3] Rojas, R., Knipping, L., Raffel, W.U., Friedland, G. (2001). *Elektronische Kreide: Eine Java-Multimedia-Tafel für den Präsenz- und Fernunterricht.* In Beck, Sommer (editors): Tagungsband der Learntec, Vol 2, pp. 533-539, Karlsruhe, October 2001.

[4] Rummel, N., Spada, H. (2002). *Combining worked-out examples and vicarious learning to promote the coordination of computer-mediated interdisciplinary collaboration.* Annual conference of AERA 2002. New Orleans, USA.

[5] Stella, isee systems - Website http://www.iseesystems.com.

[6] Bollen, L., Hoppe, H.U., Milrad, M., Pinkwart, N. (2002). *Collaborative Modelling in Group Learning Environments.* In Davidsen, Mollona, Diker, Langer & Rowe (eds), Proc. of System Dynamics Society, Palermo (Italy), July 2002, pp. 53.

[7] CoLab – Collaborative Laboratories for Europe, website www.co-lab.nl

[8] Jansen, M (2003). *MatchMaker - A Framework to Support Collaborative Java Applications*. In Hoppe, Verdejo & Kay (eds.): Proc. of AIED 2003, pp. 535-536. Amsterdam, IOS Press.

[9] Suthers, D. and Weiner, A. (1995). *Groupware for developing critical discussion skills*. Proc. of CSCL 1995, Bloomington, Indiana, 1995.

[10] ModellingSpace - website http://www.modellingspace.net

[11] Harrer, A., Bollen, L. (2004): Klassifizierung und Analyse von Aktionen in Modellierungswerkzeugen zur Lerner-unterstützung. In: Harrer, Martens (eds.): Modellierung als Schlüsselkonzept in ILLS, Workshopband, Marburg, 2004.

[12] Harrer, A., Bollen, L., Hoppe, H.U. (2004). *Processing and Transforming Collaborative Learning Protocols for Learner's Reflection and Tutor's Evaluation.* Submitted to Workshop AI in CSCL, ECAI conference, Valencia, 2004