What the Logs Can Tell You: Mediation to Implement Feedback in Training

David A. Maluf[*], Gio Wiederhold[**], Ali Abou-Khalil[***]

[*] Research Institute for Advanced Computer Science, Computational Science Division, NASA Ames. maluf@ptolemy.arc.nasa.gov
[**]Department of Computer Science Stanford University, Stanford
[***] Caelum Research, Computational Science Division, NASA Ames.

Abstract. The problem addressed by Mediation to Implement Feedback in Training (MIFT) is to customize the feedback from training exercises by exploiting knowledge about the training scenario, training objectives, and specific student/teacher needs. We achieve this by inserting an intelligent mediation layer into the information flow from observations collected during training exercises to the display and user interface. Knowledge about training objectives, scenarios, and tasks is maintained in the mediating layer. A designer constraint is that domain experts must be able to extend mediators by adding domain-specific knowledge that supports additional aggregations, abstractions, and views of the results of training exercises.

The MIFT mediation concept is intended to be integrated with existing military training exercise management tools and reduce the cost of developing and maintaining separate feedback and evaluation tools for every training simulator and every set of customer needs. The MIFT Architecture is designed as a set of independently reusable components which interact with each other through standardized formalisms such as the Knowledge Interchange Format (KIF) and Knowledge Query and Manipulation Language (KQML).

1 Mediation applied to military exercise management

The initial application of MIFT is the Exercise Analysis and Feedback phase of military exercise management as schematically shown in Figure 1. More precisely, the focus is on simulation-based army training exercises [1]. MIFT handles some of the information flows involved in training exercise management. The intent of MIFT is to supplement the flow of information from simulations to evaluation and review and complete a feedback loop by supplying information to plan and tailor future training exercises.

MIFT processes the data that is logged during training exercises and uses scenario information and domain knowledge to organize the data from the exercises in ways that are meaningful and useful for the Observer/Controllers (O/Cs) managing the exercises, trainees, commanders, can run at any location that supports Web browsing; the user does not have to download the simulation data. An innovation of the user interface is that it is designed to display information received from a mediator. Users connect to MIFT and the underlying exercise results by using a Java-capable browser. Building the user interface in a browser has several advantages:

1. Users can access exercise results in the same way they access other information from local and remote sources. The user interface will be increasingly familiar to O/Cs and trainees.
2. The exercise data may be local or remote. Startup and initialization is simple. Users do not have to download and manage the exercise data.

A key benefit of mediators for military training applications is that they avoid the need for each simulation program having to build from scratch and maintain a separate set of analysis and feedback software packages.

The operations referenced by the mediator can be layered in the direction of the data-to-knowledge aggregation as shown in Figure 2. For example, the first two levels in the mediator perform standard aggregations, selections, and analyses on the data sources. We have implemented these two levels to provide a basic level of functionality for higher levels. The third level in the mediator uses knowledge of the training scenario so that O/Cs and trainees can obtain feedback about how well specific scenario tasks have been performed. The mediator allows users to obtain specific feedback without having to understand the structure of the underlying data. A planned fourth level in the mediator will use domain-specific models about the exercise, the scenario, and causal relationships in the exercise to analyze the data for its probable significance and automatically call the users' attention to what it perceives as the more relevant exercise results. It is useful to think of the mediator as composed of three parts:

1. Data from disparate sources are converted into object instances over which inferences can be performed.
2. Knowledge about the application domain is maintained in declarative representations.
3. An inference engine processes the knowledge and data sources to produce higher level information that is passed to other mediators or to the user interface in a standardized form.

One of the MIFT functionalities is that an Observer/Controller (O/C) will depend upon it during an After Action Review (AAR) or that a

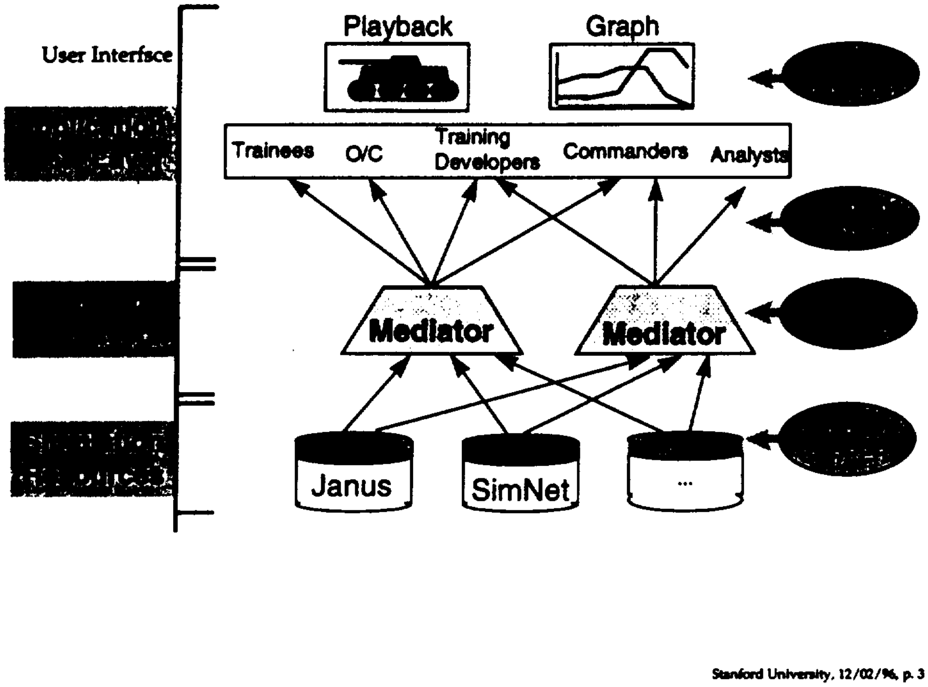

Fig. 2. The operations envisaged from the mediator can be layered in the direction of the data-to-knowledge aggregation.

trainee will use it after the AAR. As similar MIFT functionality will be useful to commanders, exercise evaluators, weapons designers, and others, but each of these other users is likely to want a different user interface and additional mediator functionality.

MIFT uses wrappers to isolate the mediators from the specific data formats and other differences between simulator outputs. When a mediator needs additional information, it calls the appropriate wrapper. The wrapper accesses the data and creates instances of the appropriate objects. The current implementation includes wrappers that process the outputs of Janus simulation runs, and LEAF[1] formated data from SimNet results. We believe that MIFT functionality can be made available

[1] Janus simulation databases.

for additional simulators by writing the appropriate wrapper to process simulator outputs. Writing additional wrappers requires programming expertise, but it is not a major undertaking. Using MIFT on a different simulation may also require additional modules and/or user interfaces to provide new functionality appropriate for that simulation. For example, the mediator that creates force ratios is more useful for simulations at the battalion or higher level and might not have been developed for analysis of simulations at the company level.

3.1 Implementing a Programmable Mediator

The architecture used for the MIFT mediator was based on a system that can sustain minimal first order logic inference capability. To further minimize development cost, the Mediator is finally written in parts in Clips [7], a widely-available and easily portable expert system shell. With little and careful programming, Clips was capable of supporting networking [2], a forward or backward chainer, a unifier, an in-memory object oriented database and a knowledge base that accepts and translate knowledge in the form of objects, rules and facts. The major function of the architecture is to allow a temporal hyper-graph construction that triggers modules to it which will perform their assigned tasks. The major five modules were as follows: A *Conflict Resolver* to maintain the truth values in the system, *Domain Modules* or the processes revolving around the domain knowledge of the main requirements, a *Report Agent* in which reports are generated and wrapped in KQML after the main requirements are accomplished, a *Maintenance Module* once some processes have terminated after the data and finally *Data Wrappers* which perform the necessary wrapping to maintain a correct syntax for the language in use. Hence, template structures are not violated. This reduces tremendously the amount of data to be loaded in comparision to the amount that will be used. Typically most databases are collections of instance events which have a time stamp associated with them and hence the wrappers are capable of playing back the databases as a function of time. Wrappers are mostly written in C++ to suit the variety and embedded complexity of the original databases.

Programming the MIFT mediator as a reusable system from task to task is performed by changing the domain module. Although attempt was made to make the conflict resolver generic in its functionalities among the tasks, domain specific rules are used in the module. The major goal of the conflict resolver is to identify the knowledge which might be disruptive to the overall mediator operation. The domain expert rules were divided under

1. Cyclic behavior: where asserted events result in cyclic effects in the process of inference.
2. Repetition and redundancy: where asserted events are redundant in the databases.
3. Constrained Space: where asserted events who's truth value conflicts with prior asserted events. For example a stated destroyed tank appearing later on in the simulation as a functional unit. Conflicts were generically sorted out using deduction rules which eliminates the erroneous event.

4 Conclusion

This paper describes Mediation to Implement Feedback in Training to customize the feedback from training exercises by exploiting knowledge about the training scenario, training objectives, and specific student/teacher needs. We plan to achieve this by inserting intelligent mediators into the information flow from observations collected during training exercises to the display and user interface functionality. Knowledge about training objectives, scenerios, and tasks is maintained in the mediators. A technical constraint is that domain experts must be able to extend mediators by adding domain-specific knowledge that supports additional aggregations, abstractions, and views of the results of training exercises.

MIFT is intended to allow analysis and evaluation software to be reused by all of the different consumers of simulation results. In addition to trainees, O/C, and commanders, others who need to analyze and evaluate simulation results include exercise planners, training managers, weapons designers, tactics developers, and doctrine writers. MIFT can also provide results to other software applications; for example, software used to assist in exercise planning and preparation can use MIFT analyses of previous exercises to identify the tasks and subtasks that need to be emphasized in additional training. Thus MIFT contributes to completing the feedback loop from the results of one simulation run into the planning and preparation for future training.

The Mediator is currently written in Clips 6.0 [7], a widely-available and easily portable expert system shell. Since user interface functions and data access functions are separated out into other components, the module implementations are quite small. For example, the force ratio computation for any set and/or combination of units is only four rules for a total of 12 lines. Most other mediators at the current stage are smaller. We believe that some domain experts will be able to write modules in Clips.

5 Acknowledgments

This research was supported by a grant from DARPA under the CAETI-EXMAN program and under contract N66001-95-C08618; Kirstie Bellman is the Program Manager. Input from Julia Loughran from IDA has been particularly helpful. This research has tremendously benefited from Ted Linden (Project Manager, Myriad Software) and Priya Panchapagesan (Stanford Graduate Research Assistant).

References

1. M. Crissey, G. Stone, D. Briggs and M. Mollaghasemi, "Training Exercise Planing: Leveraging Data and Technologies"; Proceedings of the 16th Interservice/Industry Training System and Education Conference. Washington, DC: national Security Industrial Association.
2. Clips Knowledge Networking Protocol `www-db.stanford.edu/~maluf/cknp/`
3. T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, J. McGuire, S. Shapiro, and C. Beck, "Specification of the KQML Agent-Communication Language"; Stanford February 1994.
4. M. Genesereth."Knowledge Interchange Format"; Technical Report Logic-92-1, Stanford University, 1992.
5. D. Maluf, G. Wiederhold, T. Linden and P. Panchapagesan, "Mediation to Implement Feedback in Training"; CrossTalk: Journal of Defense Software Engineering Software Technology Support Center, Department of Defense, 1997.
6. MIFT Home page `www-db.stanford.edu/mift`
7. Riley, G. "CLIPS: An Expert System Building Tool"; Proceedings of the Technology 2001 Conference, San Jose, CA, December 1991.
8. G. Wiederhold, "Mediators in the Architecture of Future Information Systems"; IEEE Computer, March 1992, pages 38-49.
9. Y. Papakonstantinou, H. Garcia-Molina and J. Widom, "Object Exchange Across Heterogeneous Information Sources"; International Conference on Data Engineering, 1995.

This article was processed using the LaTeX macro package with LLNCS style

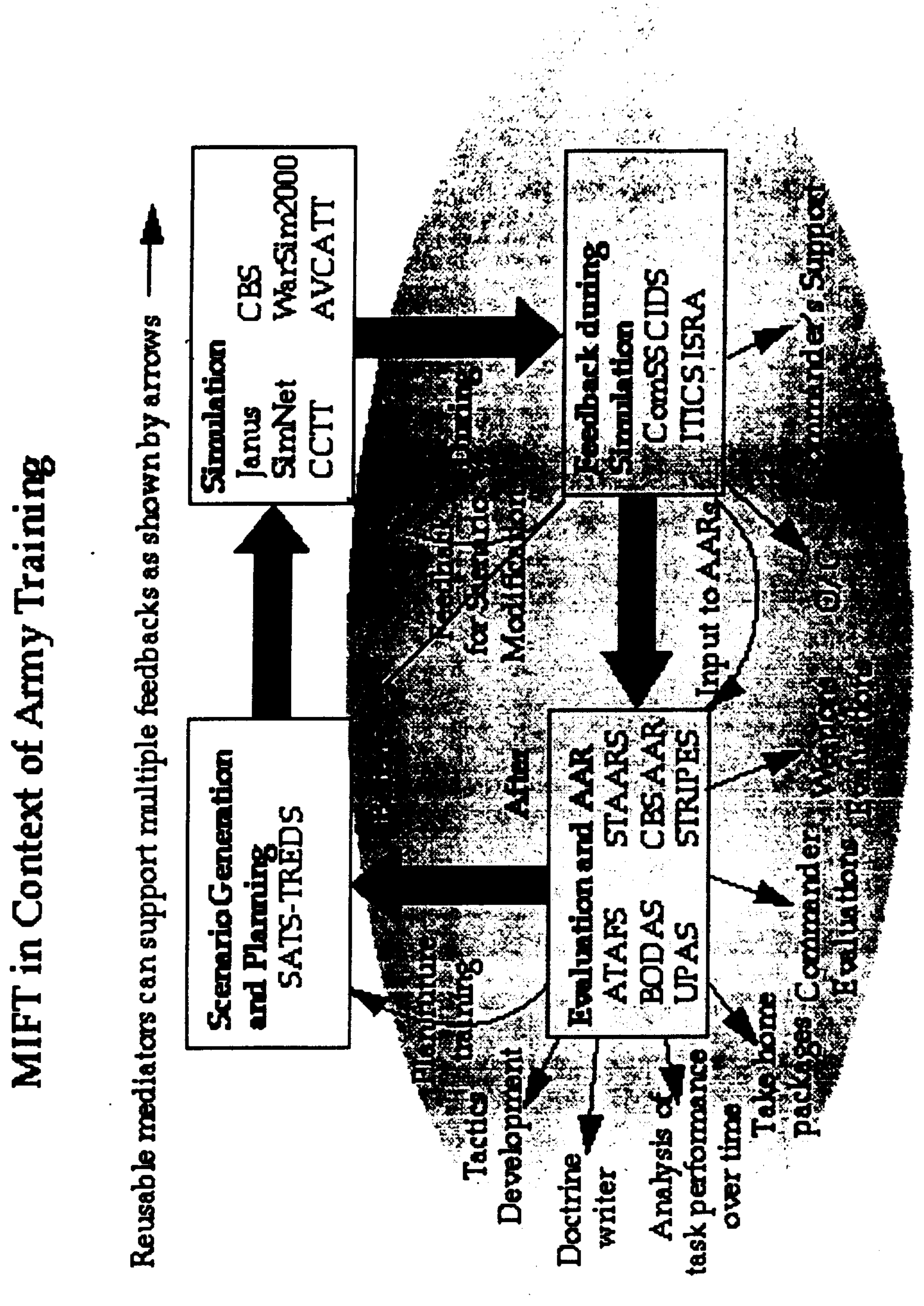

Fig. 3. The application Mediation to Implement Feedback in Training (MIFT) is the Exercise Analysis and Feedback phase of military exercise management. This figure illustrates the many different simulation results and the roles that MIFT can play by implementing reusable mediators that aggregate, summarize, and analyze simulation results and deliver them to various consumers in terms tailored to their individual needs.