Using Process Definitions to Drive User Interactions with Digital Government Systems

Lori Clarke[1], Alan Gaitenby[1], Ethan Katsh[1], Matthew Marzilli[1], Leon Osterweil[1], Daniel Rainey[2], Borislava Simidchieva[1], Norman Sondheimer[1], Leah Wing[1], Alexander Wise[1]

1. University of Massachusetts Amherst, Amherst, MA 01003,
1-413-545-2186, ljo@cs.umass.edu

2. National Mediation Board, 1301 K Street, N.W., Suite 250 East, Washington, D.C. 20005,
1-202-692-5051, Rainey@nmb.gov

ABSTRACT
This poster is showing that process definitions can be effective in specifying and guiding Online Dispute Resolution (ODR) to facilitate effective interactions and mediation among disputants. Simply specifying these interactions is not sufficient to assure that the mediator and the participants are effectively engaged in performing their roles in such processes. Our project is developing a system to translate process definition specifications into actions that are reflected in the screens that participants and mediators view while participating in these processes. Thus we are demonstrating that, with the help of the system we are building, the process definition can control changes to the user interfaces of the participants involved in the process itself. This should enable the process to keep all users continually informed about the state of the process in which they are participating including the actions that the user needs to perform. This should improve the ability of the process to be an effective aid to ODR mediation.

Categories and Subject Descriptors
K.4.3 [**Organizational Impacts**]: User interfaces, process definition, process driven interfaces

General Terms
Process, Dispute Resolution, Prototyping

Keywords
Process Definition, Process Driven Interfaces, Online Dispute Resolution

1. INTRODUCTION
The goal of Online Dispute Resolution (ODR) is to make effective use of computer and communications technologies to facilitate the mediation of disputes[1]. The appropriate application of technology, moreover, should enable new forms of dispute resolution, for example supporting negotiations where the parties are separated should subsequently lead to greater acceptance of them. We believe, moreover, that these definitions can also be sufficiently detailed that they also comprise a clear specification of just how computer and communications technologies can be used for their facilitation[2]. For this approach to succeed, however, the processes that have been automated must make it clear and easy for human participants (i.e., the disputants and mediators) to understand what is required of them and how to become active and successful in their participation. Thus, a mechanism is needed to translate process-defined activities into instructions, directives, and prompts that are clear and inviting to the participants. While the design of screen images and icons is clearly important, in this work, we focus on the equally essential job of using the process definition itself to prescribe how and when the user interface is to change to reflect progress in the process. Process definition elements are being used to dictate precisely how and when these changes are to be made and how users can be asked for needed input and can be informed of the effects of their participation and the actions of others.

2. PROCESS PROGRAMMING
The work of this project is based upon the use of the Little-JIL process definition language[3]. Little-JIL is a graphical notation that facilitates the clear communication of process ideas, including the more elusive, but no less important, concepts of exception management, resource utilization, and artifact flow. In addition, Little-JIL has a formal semantics, based upon finite-state machines, and this allows Little-JIL definitions to be the basis for rigorous analysis. Little-JIL has been used to define processes in such diverse domains as software development, medical procedure, and digital government. For this project, Little-JIL has been used to define the high level processes that the US National Mediation Board (NMB) uses to facilitate the resolution of disputes. But Little-JIL has also been used to define the lower level details of the "Brainstorming" subprocess that is at the very heart of the processes that NMB uses in dispute resolution. These experiences have confirmed that a language, such as Little-JIL, can indeed be effective in outlining the high level architecture of key processes such as ODR. Our experiences have also suggested that Little-JIL can also be effective in nailing down the details of Brainstorming processes sufficiently precisely to define the roles that computers and communication technologies can play in supporting these processes[4]. That being the case, the project is now moving forward to the creation of systems that NMB can use to exploit such technologies to facilitate their mediation efforts. In order to do this, however, these technologies must be effective in supporting the efforts of human participants. This requires the development of effective human interfaces to these processes.

3. GUI PROTOTYPING
In a previous phase of this effort, we developed a system, STORM, which presented user interface screens to participants, prompted them for inputs, accepted their inputs, and presented summaries to both mediators and disputants. The STORM prototype intentionally did not enforce process restrictions, but enabled the mediator to simulate different processes and the restrictions that they could potentially enforce on disputants.

Experience in using the STORM prototype has suggested that the screens we developed can be adequately engaging and also suggested the desirability of certain forms of user engagement and interaction. These findings have now been reflected and incorporated into more detailed, precise, and articulate definitions of ODR processes. Thus, the next phase of our project entails using these process definitions to drive user interactions with the process, and thus enforce defined process restrictions.

4. PROCESS-DRIVEN USER INTERFACES

We are now developing a glueware/middleware system that interacts with a Little-JIL process definition on one side and with the screens of the various process participants on the other side. Our new system acts as a dispatcher that can receive events from an executing Little-JIL-defined process and interpret them as directives to the user interface (e.g., changing the screen, popping up a window, graying out a button, or putting text into a text box). It also receive reports of user interactions and translates them into events that are then sent to the executing process to be interpreted as directives that are needed to drive process execution forward (see Fig.1)
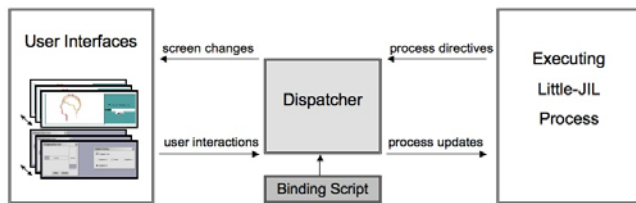

Fig 1. High-level design of the process-driven user interface and Little-JIL communication architecture.

A binding script, written by someone who has a deep understanding of both the process definition and the user interface design, describes how changes in one are to be reflected in the other. These changes are described using an abstract set of operations that can be mapped to various user interface frameworks or process state management components. To date, we have done this mapping for the Tapestry web development framework [5] and Swing [6] on the user interface side, and for the Little-JIL agenda management system on the process side [7].

This system is being used it to manage the interfaces of the mediator and the disputants in various ODR processes of interest to the NMB. With this system, it is relatively easy to change the process definition or to change the user-interfaces associated with a process definition and, of course, to maintain the correspondence among them. For example, it is relatively

straightforward to vary how much information is revealed to disputants or how much control a mediator should have over the ensuing process. This support for experimentation will allow us to define and evaluate a range of ODR processes.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] Katsh, E. and Rifkin, J., *Online Dispute Resolution: Resolving Disputes in Cyberspace*. Jossey-Bass, San Francisco, CA, 2001.

[2] Osterweil, L.J., Schweik, C.M., Sondheimer, N.K., and Thomas, C., Analyzing Processes for E-Government Development: The Emergence of Process Modeling Languages. *Journal of E-Government,* 1, 4 (2005), 63-87.

[3] Cass, A.G., Lerner, B.S., McCall, E.K., Osterweil, L.J., Sutton Jr., S.M., and Wise, A. Little-JIL/Juliette: A Process Definition Language and Interpreter. In *Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000)* (Limerick, Ireland, June, 2000).754-757.

[4] Katsh, E., Osterweil, L.J., Rainey, D., and Sondheimer, N.K. Experimental Application of Process Technology to the Creation and Adoption of Online Dispute Resolution. In *Proceedings of DG.o2006: The 7th Annual International Conference on Digital Government Research* (San Diego, CA, 2006). 272-274.

[5] Apache Software Foundation. Tapestry Framework http://tapestry.apache.org/

[6] Robinson, M., and Vorobiev, P., *Swing, Second Edition*. Manning Publications, Greenwich, CT, 2003.

[7] McCall, E.K., Clarke, L.A., and Osterweil, L.J., An Adaptable Generation Approach to Agenda Management. In *Proceedings of the 20th International Conference on Software Engineering (ICSE 1998)* (Kyoto, Japan, April, 1998). 282-291.