

Trust Resource Management in Digital Government Through Process Modeling

Lee Osterweil, Norman Sondheimer, Anthony Butterfield, Lori Clarke, Robert Marx,
Matthew Billmers, Joel Sieh, Bruce Southard, David Su
University of Massachusetts Amherst Amherst, MA 01003
ljo@cs.umass.edu; sondheimer@som.umass.edu

Abstract

This paper explores the use of process technology to create formal process models to increase the level of trust that stakeholders have in digital government. Digital Government services have been painfully slow to emerge. The paper's focus is on the digital government processes, the complex interplays of the human, institutional, and data processing entities that these systems support. Our approach is to provide mechanisms to increase trust and speed development by facilitating transparency in the development, representation and analysis of digital government processes.

1. The Challenge: Achieving Trust through Process Modeling

We are exploring the use of process technology to create formal process models to increase the level of trust that stakeholders have in digital government. The Oxford English Dictionary (OED) defines "trust" as: "confidence in some quality or attribute of a person or thing." Effective management of this trust resource must be central to digital government development, as it is elsewhere (Rousseau et al., 1998).

Digital Government services, such as law enforcement, licensing, education, tax collection, and disaster response, range in importance from minimizing annoyance to saving lives. Yet their emergence has been painfully slow, especially as compared to the rapid development of digital commerce. One typical example of the difficulty of implementing a digital government capability is that of the U.S. Customs and ITDS (Fountain 2001). Fountain chronicles attempts at realizing a new, efficient data system that would integrate U.S. Customs, Transportation, and Immigration. The system would have addressed the complexities of the North American Free Trade Agreement (NAFTA) and had the potential to save U.S. businesses billions of dollars. It appeared to be technically achievable. Yet instead, a fifteen-year-old legacy system was retained. The inability to satisfy all the concerns of agency, business, and political stakeholders stymied the effort. There are many other examples.

The focus of our work is not on the computer systems, but on the digital government processes, the complex interplays of the human, institutional, and data processing entities that these systems support. Previously, we have explored the difficult question of determining whether or not a system can be trusted to perform as desired. Our current work delves more deeply, exploring the question of whether or not the processes that the system supports and implements are themselves trustworthy. The goal of our effort is the exploration of how to use specific technologies as vehicles for enhancing the trust of government and non-government stakeholders. Our approach is to provide mechanisms to increase trust by facilitating transparency in the development, representation and analysis of digital government processes.

2. Our Approach: Trust Resource Management through Process Technology

Our multidisciplinary effort brings computer and information science software engineering together with social science research in organizational behavior methods, to utilize process technology as the foundation for what we call **Trust Resource Management (TRM)**. We are exploring the language needed for effective and efficient definition of complex digital government processes; the capabilities required for analyzers and interpreters of these languages to demonstrate the presence or absence of process properties critical to the interests of digital government stakeholders; and the process development methods that efficiently interpret stakeholder needs, create process models, demonstrate their properties and build stakeholder trust.

Of fundamental importance, but surprising difficulty, is the ability to describe government processes accurately. Government processes involve complex systems of people, organizations, computational systems and legal constraints. Application for a driver's license may appear simple when the applicant

encounters little difficulty, but the process must allow for dozens of special considerations such as clerical errors, withdrawal and resubmission of the application, and handling issues relating to payment. Worse, most existing processes are defined in natural language, if they are defined at all. This leaves open room for misinterpretation that can lead to process inefficiencies and errors, which in turn can destroy trust. Because of this, a considerable amount of research has been focused on the development of languages and formalisms to define processes completely, precisely, and clearly (Katayama, 1989; Kaiser et al., 1993). This work has served to emphasize both the difficulty and the importance of research in this area.

We studied in earlier work the concept that processes are a form of software amenable to programming using appropriate languages (Osterweil, 1987; Osterweil, 1997). We suggested that processes can and must be defined precisely, clearly, and completely using representation approaches that are based upon programming languages. In our view these process definitions should be developed in response to process requirements that originate with the various process stakeholders. Our success with this research, as applied in other domains, suggests that these approaches can be of substantial value in Trust Resource Management. This work leads us to our first hypothesis.

Hypothesis 1 (Language Hypothesis): Digital Government processes can be represented clearly, completely, and precisely through rigorously defined modeling formalisms.

Social, behavioral and political science research all have large bodies of literature that identify Critical-To-Trust properties (Fountain, 2001; Wilson, 2000). These needs are expressed in very specific terms. For example, the development of one or more systems to support a new occupational therapist (OT) license look-up and renewal process. The OT professional society could well require that addresses be shown only under agreement of the individual OT. The IT Department might require that the system not need support from a new type of database management system. The state portal call center staff may require that the process enforce a fixed amount of advance notice of changes in renewal requirements. Each group will want to verify that the new process addresses their requirements before proceeding to the development of systems to support the process.

We are applying two complementary approaches: dynamic testing and static analysis (Osterweil, 1996). Dynamic testing monitors process execution and is effective in detecting deviations from desired or mandated behaviors. Static analysis is aimed at demonstrating the absence of defects and undesirable behaviors, but only for limited classes of defects. In the next stage of our work, we will evaluate both dynamic testing and static analysis approaches to understand their relative effectiveness in enhancing trust. Therefore, our second hypothesis is:

Hypothesis 2 (Analysis Hypothesis): Many Critical-To-Trust properties of digital government processes can be assessed objectively and definitively through the application of rigorous analyzers to appropriate process representations.

The complexities of designing and implementing digital processes in the industrial sector are amplified in the governmental sector. The number and variety of stakeholders, political considerations, and budgetary vagaries all add to conflict, competition, and distrust. Stakeholders must be involved in developing the requirements for government processes to avoid these problems.

Increasing the trust of stakeholders in digital government processes requires a redesign of the leadership, teamwork, and group dynamics associated with the development of such processes. Our research in team training, decision improvement, trust analysis, and relapse prevention encourage us to integrate the languages and analyzers of Hypotheses 1 and 2 into a participatory process development method. The key will be to decide how to engage stakeholders in making the key process development decisions. We are basing our work on the concepts of Joint Application Design and Participatory Design (Wood and Silver, 1995) and the decision-making rules using methods developed by Victor Vroom (Vroom et al., 1998). This leads us to our final hypothesis.

Hypothesis 3 (Development Hypothesis): Participatory development methods can facilitate the determination of Critical-To-Trust digital government processes and properties, and with process languages and analyzers, enhance stakeholder trust in digital government processes.

These three hypotheses define our proposed method. Participatory process requirements meetings will bring key stakeholders together. These stakeholders will specify the Digital Government process requirements and define the properties these processes must have in order to engender trust. Process

definers will then develop precise process definitions for study by automated analyzers that will then determine if the process definitions do indeed satisfy these properties. If not, the process, its requirements, or the critical-to-trust properties must be modified. As the Digital Government process evolves, the process definition, its requirements and its properties will evolve with it, and the analyzers will be reapplied.

3. Progress to date: A Process Requirements Specification Field Study

The United States Commonwealth of Massachusetts Office of Consumer Affairs and Business Regulation (OCA) is responsible for the administration of hundreds of various types of licenses, permits and registrations (henceforth called “licenses”). OCA is chartered to develop an online license renewal (OLR) system implementing renewal processes that must be trusted by a variety of stakeholders.

The authors acted in a consulting capacity to OCA in reviewing and assessing agency renewal process requirements as part of the Commonwealth of Massachusetts’ OLR project requirements gathering phase. OCA chose the UML Use Case formalism as its vehicle for process requirements specification. Use Cases use natural language (English) as they attempt to capture the required flow of a process, as well as exceptional behavior and other requirements concerns (Kulak et al., 2000).

UMass selected a representative set of those Use Cases and specified them using our process modeling language, Little-JIL (Cass et al., 2000), in order to evaluate the use of a process language in the development of process requirements. Our experience was positive, as our work revealed many points of concern in the requirements as specified by the Use Case formalism. We focused on a total of seven licensing processes belonging to two agencies. The sample was then expanded, providing further areas to identify and highlight possible concerns. We then reviewed 19 more of the 43 Use Cases provided and classified the irregularities we found into three groups. These failings were likely to lead to development of systems whose characteristics and behaviors would be likely to be unpredictable, and presumably less worthy of stakeholder trust. By clearing up these failings, we believe that our use of a more rigorous process formalism will lead to increased trust in the process.

A short summary of some of specific findings follows (see Sondheimer et al., 2002 for more detail):

Actor Structure: We found five actor inconsistencies during our investigation of the Use Cases. Looser control of actor specification in Use Cases allowed the requirements engineers to omit actors unintentionally. Little-JIL was effective in identifying these problems during the creation of the process requirements, and whenever they were changed. The Little-JIL formalism defines process flow and coordination through the use of a step abstraction that requires an actor binding for each step. This prevented the accidental omission of these actors, and focused attention on specifying them consistently.

Parameter Flow: We found seven errors in Use Cases caused primarily by lack of parameter flow information. Little-JIL includes a type-safe parameter flow system. This requires process definers to clearly define what data are required for the completion of each step, what data each step produces, and how the data flow among the steps. This additional rigor helped us to detect data flow inconsistencies that may have gone unnoticed in the processes whose requirements were being defined.

Lack of Process Component Reuse: We found three significant redundancy issues in the Use Cases. Use Cases are basically lists of steps written in natural language. Finding commonalities and abstracting the commonalities to a higher level is inefficient and difficult. Missing these commonalities can lead to errors, inconsistencies, and generally eccentric behaviors as the requirements specification undergoes changes. Little-JIL addresses this problem because it is a hierarchically structured process language. It encourages process definers to identify logical clusters of related process steps and then group those steps together under one higher-level step. This increases readability and allows for reuse of steps or groups of steps from any level of the process.

The problems identified and corrected in our experimentation demonstrate that using an appropriate process specification language to specify process requirements can help prevent important errors that the Use Case formalism is less likely to prevent. Formalisms with richer semantics than the UML Use Cases are useful for crafting clear, precise, and complete Digital Government process requirements specifications. Little-JIL documents may be larger than Use Case documents and may take longer to specify. We believe this is because of the increased rigor required, but consider the benefits in clarity and completeness, leading to increased process trustworthiness, to be adequate compensation.

4. Future Plans

This brief study of the use of superior formalisms to support process requirement specification must be one of many before we can argue that our three hypotheses are correct. Our future work will center on the full development of prototype digital government processes, their evaluation within the context of the Commonwealth of Massachusetts digital government portal, mass.gov, and the use of these evaluations to improve the process definition, analysis, and development capabilities, as well as the processes themselves. The work will necessitate the iteration of a loop consisting of government process requirements specification, property specification, implementation, evaluation of results, and improvement of both processes and process technology. The evaluation to be carried out will be essentially evaluation of our three hypotheses. The Language Hypothesis will be evaluated through the application of Little-JIL to process definitions. The Analysis Hypothesis will be evaluated through the application of the various dynamic and static analyzers to Little-JIL process definitions. The Development Hypothesis will be evaluated by embedding process technology initially in JAD, facilitating digital government creation and observation of the general methods by which these technologies are used to assure the steady incremental development and improvement of digital government processes.

The next stage of our research will enhance trust through the realization of digital government by incorporating the disciplines outlined here: That rigorous process languages, analyzers, and participatory development methods can be combined to deliver enhanced stakeholder trust in digital government.

5. Acknowledgements

This material is based upon work supported by funds from the Commonwealth of Massachusetts' Information Technology Division and the National Science Foundation under Grant No. EIA-223599. We would like to thank Val Asbedian, Tim Healy and Tom Price for supporting and helping supervise the work from the Government side and Annie Kelly for editorial support. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the Commonwealth of Massachusetts or the National Science Foundation.

6. References

- Cass, A. G. et al., 2000. Little-JIL/Juliette: A Process Definition Language and Interpreter. *International Conference on Software Engineering*, Limerick, Ireland, pp. 754-758.
- Fountain, J., 2001. *Building the Virtual State: Information Technology and Institutional Change*. The Brookings Institution, Washington, DC.
- Kaiser, G. E. et al., 1993. A Bi-Level Language for Software Process Modeling. *15th International Conference on Software Engineering*, Baltimore, MD, pp. 132-143.
- Katayama, T., 1989. A Hierarchical and Functional Software Process Description and its Enaction. *11th International Conference on Software Engineering*, Pittsburgh, PA, pp. 343-353.
- Kulak, D. et al, 2000. *Use Cases: Requirements in Context*. Addison Wesley Professional, NYC, NY.
- Osterweil, L. J., 1987. Software Processes are Software, Too. *Ninth International Conference on Software Engineering*, Monterey, CA, pp. 2-13.
- Osterweil, L. J., 1996. Perpetually Testing Software. *Ninth International Software Quality Week*, San Francisco, CA.
- Osterweil, L. J., 1997. Software Processes Are Software, Too, Revisited. *19th International Conference on Software Engineering*, Boston, MA, pp. 540-558.
- Rousseau, D.M. et al., 1998. Not So Different After All: A Cross-Discipline View of Trust. *The Academy of Management Review*. **Vol. 23, Num. 3, pp. 393-404.**
- Sondheimer, N.K., et al. *Final Report: Application of Process Modeling to Online License Renewal*. 2002, Electronic Enterprises Institute, University of Massachusetts, Amherst, MA.
- Vroom, V.H., et al., Participative Leadership, in *Leadership: The Multiple-Level Approaches: Classical and New Wave*, F. Dansereau and F.J. Yammarino, Eds. 1998, JAI Press: Stamford, CT. pp. 145-189.
- Wilson, J., 2000. *Bureaucracy: What Government Agencies Do and Why They Do It*. Basic Books, NYC, NY.
- Wood, J. and Silver, D., 1995. *Joint Application Development*. Jossey-Bass, San Francisco, CA.