Ubiquitous Process Engineering: Applying Software Process Technology to Other Domains

Leon J. Osterweil
Laboratory for Advanced Software Engineering Research
Department of Computer Science
University of Massachusetts
Amherst, MA 01002
USA

Abstract

Software engineering has learned a great deal about how to create clear and precise process definitions, and how to use them to improve final software products. This paper suggests that this knowledge can also be applied to good effect in many other domains where effective application of process technology can lead to superior products and outcomes. The paper offers medical practice and government as two examples of such domains, and indicates how process technology, first developed for application to software development, is being applied with notable success in those areas of endeavor. The paper also notes that some characteristics of these domains are highlighting ways in which current process technology seems to be inadequate, thereby suggesting ways in which this research is adding to the agenda for research in software process.

1.0 **Introduction**

In earlier papers we have argued that the processes that are used to develop software should themselves be viewed as software [6, 7]. We have suggested that software processes should themselves be developed by means of careful processes, starting with requirements specification, proceeding through architecture and design, and then defined precisely with languages that have execution semantics. We have argued that the execution of such *process programs* can effect the superior coordination of agents, both human and automated [7]. In more recent work we have also demonstrated that there can be considerable value in taking such executable processes as the subjects of dynamic analyses such as simulation runs [Bin ProSIM paper], and as the subjects of static defect analyses [Raunak ProSIM, [1]. Our view is that these analyses are essential components in continuous process improvement loops that 1) baseline existing processes, 2) use analysis to identify defects and shortcomings in these processes, 3) propose improvements to the processes, 4) use analysis to verify the value of the proposed improvements, and 5) deploy the improved processes, thereby creating the basis for the next improvement iteration. The ultimate goal of all of this is the establishment of a discipline of software process engineering in which the continuous enhancement of processes leads to increasing quality in final products, and increasing efficiency in producing them. In a previous paper [Osterweil keynote 05] we have argued that more precise and comprehensive process definitions are more effective bases for the kinds of definitive analyses that lead more efficiently to successful improvement efforts. We referred to research aimed at such more precise and definitive languages and analyses as *microprocess* research.

Concurrently with these increasing understandings of the possible mechanics of continuous software process improvement, and the particular value of microprocess research, has come the realization that software development is only one of a multitude of diverse domains in which the continuous improvement of processes can lead to important benefits. We observe that processes are found universally in our society, and indeed that there are many domains in which the role of processes is at least as central as is the case in software engineering. Most of the work of government, for example, is essentially the creation and execution of processes. Likewise, medical care is centered importantly upon the devising and careful execution of processes. Processes are also at the core of such other domains as business, engineering, finance, dispute resolution, and law. Many other example domains can be identified quite readily. This observation immediately suggests, therefore, that the software process technologies that our community has developed have much to offer these other disciplines. Significantly, our preliminary investigations of these other areas has suggested that the notions of process in those domains are far less well developed than they are in software engineering, and that the rigor that we increasingly employ to good effect in software process engineering is lacking in process approaches in these other areas. Thus our current view is that these other domains have much to gain from the application of technologies that we have developed in software process engineering.

This paper explores that premise, indicating the sorts of process issues that we have encountered in other problem domains, and the ways in which we are finding software process technology to be relevant and effective. We also note that certain shortcomings of our process technologies have been highlighted in attempting to apply them in these other domains, suggesting areas for further research.

2. **Goals and Motivations for Process Research:**

To understand the relevance of software process technology to other domains, it helps to note that the goals of all of these communities have quite a lot in common. In software process research, there is a very wide range of specific technologies and approaches, but the main goals are far smaller in number. These goals, some of which are elaborated upon briefly below, are generally shared by many other domains, as shall be indicated. This suggests that technologies that have been successfully applied in our domain of software development, should have relevance in the others. Here are some of the key goals of process technology.

Team Coordination: Much of the most difficult and challenging work in virtually all sectors of society requires synergistic collaboration among many diverse contributors. In software development, a product requires careful coordination among designers, coders, testers, managers, documentation specialists, and clerical personnel. In government, such activities as licensing, running elections, and legislation itself, require a similar sort of collaboration among bureaucrats, citizen volunteers, boards, panels, and elected officials. In both domains, a clear and precise specification of the processes to be carried out can improve the chances that all members of the team have the same view of what they should be doing, and how they are to interact with each other.

Efficiency Improvement: An immediate corollary of the previous goal is that improved understanding of the ways in which teams should coordinate their efforts facilitates the identification of bottlenecks and other obstacles to improved team efficiency. In emergency room medical care this is a particularly serious problem, as delays are pervasive, and can cause pain and loss of life. Identification of bottlenecks and inefficiencies, based upon a clear specification of how agents must coordinate their efforts can be used to identify where process changes, or resource reallocations can speed processing, and increase productivity.

Automated Support: Carrying the previous point further, we note that the performers in such processes need not always be humans. In software development, compilers, design tools, test automation tools, etc. can play useful roles. In medicine, medication dispensers, infusion pumps, medical records databases, and patient monitoring devices play similar roles. In both cases, an executable process definition can be used to delineate the exact roles of such automated devices, and to provide APIs and other interfaces that enable such automation aids to be coordinated with human activities. The executable process can then be used as an integration structure for the insertion (and incremental growth) of automation.

Education and Training: Process technology can be particularly useful in supporting the training and education of the humans who will participate in processes. Most large software organizations require that newcomers acquire indoctrination in the methods that their new organizations use to build software. In a similar way, commercial enterprises, such as banks and manufacturers also require such training to assure that new employees can become proficient at their jobs more rapidly. Process simulation technologies are an example of an approach that can provide strong support in this area.

Continuous Improvement: Highly visible activities such as the CMM [3], CMMI, and ISO 9000 [SE textbooks] projects have emphasized the key role of process in effecting continuous improvement in software development. Other areas, such as manufacturing, have made similar observations. Indeed the use of such institutions as quality circles has resulted in impressive gains in quality and productivity in industry. In both cases, strong understandings of the current process, precise understandings of proposed improvements, and effective devices for measuring results are some keys to success in process improvement. A particularly key aspect of such improvement efforts seems to be effective analysis of both the current, and the proposed new, processes. Analysis that identifies defects is needed to trigger improvement, and analysis aimed at demonstrating that new processes actually achieve desired improvements (without creating new problems) is needed to carry the improvement processes further.

Reinvention: It has been suggested (eg. in [Jeffrey 05 keynote]) that, while steady incremental improvement is useful in software development, there must also be a place periodically for radical change, or reinvention. This is certainly also true in such areas as dispute resolution, where face-to-face negotiation is increasingly giving way to computer-mediated Online Dispute Resolution (ODR). In both domains, radical change can happen only after a firm understanding of the essentials of processes can be distilled.

Such deep understandings are facilitated by the use of powerful formalisms, employing appropriate abstractions to suppress superficial detail and accentuate deeper conceptual issues. Thus, pursuit of formal process abstractions is useful in helping various domains seek and achieve radical change.

The foregoing has convinced us that the overarching goals that software process research seeks to address with its technologies are goals that are shared by medicine, government, business, and a wide range of other human endeavors. Thus we embarked upon a program of research aimed at determining the extent to which this is correct, attempting to apply software process technologies to specific problems in these other domains. In the following sections, we indicate how technologies originally developed for software process are now being employed in other domains where processes are also of central importance. In some of these domains, as had once been the case in software development, the centrality of process issues had been largely overlooked. But, with growing recognition of the centrality of process issues, the potential for important and fundamental contributions of process technology is growing rapidly.

3.0 Medical Safety:

A highly influential report from the US Institute of Medicine [5] estimated that each year at least 97,000 people die from preventable medical errors in US hospitals. Far more suffer pain and non-lethal damage from such errors. The costs in money are far harder to estimate, but must be measured at least in hundreds of billions of US dollars each year. While there are many causes of the problems leading to these deplorable statistics, it seems clear that one of the most central is lack of control of the processes that are used to deliver medical care. A team consisting of University of Massachusetts software engineering researchers, and faculty from the School of Nursing, is working with administrators and workers at Baystate Medical Center in Springfield, Massachusetts, USA, to investigate ways in which process definition and analysis technologies might be effective in addressing medical care improvement goals. Our research team has identified three specific process areas as candidates for the application of process technology. In each area the goals are strikingly similar to software process goals, and the applicability of software process technology is proving to be correspondingly appropriate.

Emergency Room Operations: The Baystate Emergency Division is like most other Emergency Divisions in that patients must endure very long delays, and often simply leave without receiving care at all. We are investigating the use of process definition and simulation to determine how they might be used to improve efficiency in emergency room operations. Clearly scarcity of such resources as beds and physicians contribute to waiting. But process inefficiencies seem to exacerbate this situation. We are using process simulations to explore how different resources mixes (eg. providing additional doctors and/or nurses) can be expected to reduce waiting times and increase overall capacity to provide services, especially in response to disasters. We are also using these definitions and simulations to support the exploration of radical change in the operation of the Emergency Division. Languages and simulators originally intended for supporting software process studies are proving to be very appropriate and effective in this work, facilitating consideration not only of resource reallocation schemes, but also reinvention of the entire range of Emergency Division operations.

This research is, however, underscoring the difficulties in developing accurate definitions of highly concurrent processes. Emergency medical care is characterized by the need for promptness in acute medical situations, and the need for resources to be assigned to multiple tasks concurrently. These, and similar, reasons dictate that standard medical care procedures may need to be quite flexible, and yet must sill conform to some very basic procedural requirements. Our attempts to use software process definition languages to represent these processes are sharpening our understanding of the shortcomings of such languages in areas such as concurrency specification.

Chemotherapy Administration: We are investigating the use of precise process definition and analysis technologies to improve team coordination in chemotherapy administration, and to remove defects from the processes that those teams execute. A major focus of this work is the use of process definition and analysis techniques to identify process defects, and to incorporate continuous process improvement into the operations of the Chemotherapy Division of the Cancer Center. Chemotherapy is the process of using extremely powerful and dangerous drugs to destroy cancer cells, without causing undue harm to other cells in the body. Chemotherapy requires extremely careful measurements of the human body, and the very carefully monitored infusion of drugs over extended periods of time. The reaction of the patient's body to this regime must be monitored closely by diverse agents, and from different perspectives. Negative reactions must be noted and responded to appropriately, and interactions with other drugs that might be used to deal with other medical situations, must be continuously considered and adjusted for.

The processes for doing all of the above are complex, and require coordination among various doctors, nurses, pharmacists, clerical personnel, and the patient. One of the particularly noteworthy features of such processes is that they incorporate considerable amounts of redundancy, aimed at the prompt detection of errors and prevention of harm to the patient. Precise definition and analysis of these processes has helped the Chemotherapy Division to identify ways in which team coordination can be improved, and process defects can be detected. All of this has led to an understanding of how continuous process improvement can lead to overall benefits. One specific example is that our rigorous analysis of these processes has led to the identification of paths through the processes where some redundancy checking can potentially be bypassed, and other paths that seem to cause excessive amounts of redundancy checking.

Blood Transfusion: We are also investigating the use of process definition to improve the education and training of nurses in such processes as blood transfusion, and the use of process monitoring and analysis to reduce the incidence of such catastrophic errors as the infusion of incorrect blood types into patients. The goals here are improved team coordination, facilitation of training, detection and removal of defects, and continuous improvement. As in the cases of the other medical process projects, we are finding that much of our process technology is highly applicable. In particular, we have applied fault tree analysis to suggest such vulnerabilities as single points of failure in existing processes [Bin paper in this proceedings], suggesting ways and places were redundancy should be added.

4.0 Digital Government:

The application of software process technology to the domain of government seems no less appropriate, and our work is demonstrating that it should be quite effective here as well. In this domain, such familiar goals of teamwork improvement, efficiency enhancement, defect removal, reinvention, and automation are also of great importance.

License Renewal: In earlier work [8] we have demonstrated that precise process definition can provide important benefits in the domain of license renewal. The US state of Massachusetts, like most governmental units worldwide, is responsible for the issuance of literally hundreds of different kinds of licenses. The processes for doing this are more complex than is generally understood, typically involving the receipt of a renewal request, the search of various records and archives (eg. for the existence of complaints), the receipt of funds, and the issuance of the actual licensing materials. All of these activities must take into account the possibility of various kinds of exceptional situations. We have applied precise process definition and analysis technologies to this problem area and have identified places where process details were lacking, places where increased redundancy was desirable, and ways in which computer automation could be inserted, leading to transition strategies for digital (electronic) government.

Election Processes: We note that elections are highly complex processes, starting with voter registration activities, and continuing on through the conduct of actual elections, the accumulation of results, and the need to administer recounts and various dispute resolutions. Much attention in the US has been focused on elections since the contentious 2000 Presidential election. Most of this attention has been directed to the possibility of incorporation of electronic vote recording devices into election processes. While analysis of the soundness of these devices is clearly important, our own work emphasizes the importance of assuring the soundness of the election processes themselves, and the appropriate integration of electronic voting devices into these overall processes [Raunak paper in this proceedings]. We note that elections require the coordination of a wide range of humans, as well as automated devices. We are starting to study these processes, emphasizing the identification of defects, searching for appropriate levels of redundancy, appropriate use of electronic voting devices, and the vulnerability of election processes to frauds and collusions. In this latter work, we have used our process definition technologies to represent the behaviors of fraudulent and collusive election agents, as well as the election processes themselves. Early results of this research are contained in a paper in these proceedings [Raunak election].

Labor-Management Dispute Resolution: We have also explored the application of process technology to the domain of dispute resolution through a project involving collaboration with the US National Mediation Board (NMB). While it may initially seem that dispute resolution is an area in which rigorous process definition would be unlikely to be effective, our initial research has indicated that this is not the case. As with many other process domains, dispute resolution requires a great deal of human ingenuity and initiative, but also a great deal of discipline and structure. The NMB has long since recognized this and has over the years developed some structured approaches to bringing disputants together in productive negotiation sessions. We have applied our rigorous process definition technologies to define these processes. In doing so, it has been easier to identify ways in which technology can support these processes [4].

In addition, our work with NMB has suggested the possibility of radical change to negotiation processes. For many years, NMB negotiations were carried out primarily in face-to-face sessions. But with the growth in the number of disputes to be resolved, and the increased availability of computer and communications technologies, NMB has become increasingly interested in augmenting or replacing face-to-face negotiation session with Online Dispute Resolution (ODR) approaches. In doing so, it has become increasingly clear that ODR requires more radical change to existing processes than simple replacement of some existing process steps with automation. Our process definitions have become the basis for the radical change to ODR that NMB needs to undertake.

The NMB project has also suggested another important goal for process technology, namely the involvement of broader stakeholder groups in the specification of requirements for processes, and indeed for the automated systems imbedded in them. NMB has noted that the acceptance of the agreements reached in dispute resolution is strongly enhanced when the parties to the dispute have had an active and effective role in designing the dispute resolution processes, and in monitoring that the processes have indeed been followed. The use of a clear, yet complete and precise, formalism for defining NMB's dispute resolution processes is being explored as just such a vehicle for developing processes that are sufficiently transparent that disputants will be more receptive to acceptance of dispute resolution outcomes [paper submitted to RE 06]. We are currently engaged in work aimed at involving disputants in definition of ODR processes that will be used in resolving their disputes. Our technologies will also be used as the basis for displaying the progress through these processes to assure disputants that the agreed upon processes are indeed being followed.

5.0 Additional Domains:

As noted above, many other domains seem equally appropriate for the application of process technology. Our own research has applied these technologies to the definition of processes that should be used in creating scientific datasets that are suitable for use by researchers other than those who have created these datasets [2]. Our work has indicated that many such datasets contain data that has undergone considerable complex transformation and analysis before being published. Scientists using these datasets for their own work are well advised to be aware of the various transformations that have been applied to this data, but generally documentation of such transformations is unavailable. Early work aimed at providing that documentation has indicated that the level of precision needed in order to assure safe reuse of these datasets require the use of process definition languages that employ strong semantic power. These process technology approaches resemble in striking ways approaches previously developed and applied in software process, especially as used in configuration management.

The application of process technologies to such other domains as manufacturing, banking, management, law, and military operations seems quite promising, and indeed has been begun in some cases. Software process researchers would do well to consider the benefits derived from applying the technologies in their grasp at present to these new domains.

As noted above, these domains will benefit from such work, but consideration of the applicability of these technologies will also lead to understanding of ways in which the technologies could benefit from extensions in directions indicated by the demands of these new application domains. We have already noted that defining processes in medicine has suggested the need for stronger language features for defining concurrency. Other process language and analysis shortcomings, such as the need for superior process abstractions, the importance of clearer artifact flow definition, and the need for improved support for specification of process properties, all have been underscored by this research, suggesting new roadmaps for software process technology research.

6.0 Acknowledgments

The author wishes to express his gratitude to numerous individuals who have participated in this research, and clarified the points made in this paper. Sandy Wise, a key architect of the Little-JIL process definition language, has supported every aspect of this work. My co-investigators, Lori A. Clarke, George Avrunin, and Beth Henneman have been key leaders of the work on Medical Safety. Special thanks go to Phil Henneman and Fidela Blank for support of the Emergency Division research, and Wilson Mertens and Lucy Cassels for their strong support of our Chemotherapy process research. My co-investigators, Ethan Katsh, Dan Rainey, and Norm Sondheimer have been key leaders in the Digital Government work. Many students have also supported this work. Special thanks go to Mohammed S. Raunak, Dave Miller, Irina Ros, Rachel Smith, Matt Goetz, Bin Chen, Matt Marzilli, Natalie Podrazik, and Dan Gyllstrom.

This material is based upon work supported by the US National Science Foundation under Award Nos. CCR-0427071, CCR-0204321 and CCR-0205575. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of The National Science Foundation, or the U.S. Government.

REFERENCES

1. Clarke, L.A., Chen, Y., Avrunin, G.S., Chen, B., Cobleigh, R., Frederick, K., Henneman, E.A. and Osterweil, L.J., Process Programming to Support Medical Safety: A Case Study on Blood Transfusion. in *Proceedings of the Software Process Workshop*, (Beijing, China, 2005), Springer-Verlag Lecture Notes in Computer Science, 347-359.
2. Ellison, A.M., Osterweil, L.J., Hadley, J.L., Wise, A., Boose, E., Clarke, L.A., Foster, D., Hanson, A., Jensen, D., Kuzeja, P., Riseman, E. and Schultz, H. An Analytic Web to Support the Analysis and Synthesis of Ecological Data. *submitted to Ecology/Ecological Monographs*.
3. Humphrey, W.S. A Discipline for Software Engineering. in *SEI Series in Software Engineering*, Addison-Wesley, Reading, MA, 1995.
4. Katsh, E., Osterweil, L. and Sondheimer, N.K., Process Technology for Achieving Government Online Dispute Resolution. in *National Conference on Digital Government Research*, (Seattle, WA, 2004).
5. Kohn, L.T., Corrigan, J.M. and Donaldson, M.S. (eds.). *To Err is Human: Building a Safer Health System*. National Academy Press, Washington DC, 1999.
6. Osterweil, L.J., Software Processes are Software, Too. in *Ninth International Conference on Software Engineering*, (Monterey, CA, 1987), IEEE Computer Society Press, 2-13.
7. Osterweil, L.J., Software Processes Are Software, Too, Revisited. in *19th International Conference on Software Engineering*, (Boston, MA, 1997), 540-558.
8. Sondheimer, N.K., Osterweil, L.J., Schweik, C., Billmers, M., Canavan, D., Kelly, A., Lee-Davis, C., Li, C. and Sieh, J. Online License Renewal Analysis: Process Modeling and State Practice, Electronic Enterprise Institute and the Center for Public Policy and Administration, University of Massachusetts, Amherst, 2002.