

SESSION SUMMARY

Team Efforts

Kouichi Kishida and Dewayne E. Perry

Kishida-san and Perry presented complimentary introductions to the session: Kishida-san focusing on the important differences in the goals of various process agents, Perry focusing on the technical aspects of supporting the various process agents. The terms “inside”, “outside”, and “beside” are used metaphorically to emphasize the dichotomy between the authority and direction of management and the freedom and creativity of the individual.

Kishida-san presented four levels of concern in considering team efforts in the software process: organization, project, group, and individual. The higher levels are dominated by a view that is characterized as macroscopic, “outside”, mechanistic, and epic; the lower levels are dominated by a view that is characterized as microscopic, “inside”, humanistic, and lyric.

The house of software development consists of two floors and a basement. The second floor is populated by process designers and process managers and represents an “outside” view of development. The first floor is populated by software developers and represents an “inside” view. The basement houses the process support (that is, is the foundation) for both the developers and the managers and as such represents yet another “outside” view.

Human beings as a process agents produce both good and bad results: they do either more or less than specified. A human process agent is likely to do less when he or she is forced to be “beside” him or herself; he or she is likely to do more when he or she is free to be “inside” him or herself.

Perry’s depiction of the levels of concern was in terms of individual, family, city and state process models, with the emphasis on the last three in this context of teams. He posited three dimensions in considering team efforts: 1) inter- and intra-team issues, 2) information and interaction, and 3) the problems of scale, instantiation, customization and evolution.

Analogous to Humphrey’s 5 levels of process maturity, Perry posited 5 levels of team maturity:

1. Single-use, User-Tool, and Tool-Tool;
2. Recognition of team issues;
3. Information/Interaction support mechanisms;

4. Information/Interaction abstractions; and
5. Information/Interaction definition facilities that may be product-state based, project-state based, and/or process-state based facilities.

It should be noted that the process work at levels one and two is not “bad” work. It may well be very good work with respect to a different dimension. However, in supporting team efforts, it is immature. (Redwine offered a different characterization of team maturity in which team operations range from chaotic to smooth — Perry viewed that as more a characteristic of a particular team effort process than a characteristic of team effort modeling.)

From the position papers, the following levels were found. Level 3 information/interaction support mechanisms include triggers, attribute grammars, broadcasts/ipc/messages, (multiple) rule-chains, object relations, petri nets, wait/transaction graphs, conversations, function collaborators, hierarchical experimental databases, and software configuration management. Level 4 team abstractions include permissions, consensus, notification, and collaboration. Level 5 abstraction definition facilities include laws and policies.

The following questions were then posed for discussion. How do you support the sharing, distributing and restricting of information? How do you support the enabling, controlling, and restricting of interactions? How do you support the scaling of team efforts? How are process models customized and instantiated for teams? How is the evolution of these team models supported? What are appropriate team abstractions? What are appropriate team abstraction definition facilities?

Osterweil noted that both the authoritarian and libertarian aspects have problems. What one would like to have is a mechanism to restrict the process and be able to use it wisely. Kishida-san partly agreed: control is certainly a concern of management and is used to ensure success. Control is part of the “outside” view; and individual’s “inside” view is concerned about fulfillment and creativity, not control and success. Redwine and Diamant both noted that team members are concerned about success, the latter also arguing against unnecessary rules.

Mechanisms are needed, argued Osterweil, that enable people to feel comfortable with restrictions. For example, we need requirements and design models of the process and a transparent mechanism to view the current process state. Perry agreed. Schaefer noted that one could imagine not objecting to a manual process because it could be selectively ignored but objecting to it strenuously when it is enforced by the supporting process environment.

Tamai-san claimed that while the state of the product was important, perhaps more important is the process by which that state is reached. That process is where people are involved and where motivation, morale, and consensus are of paramount importance.

Given the fact that we do not know what the “right” process is for a particular individual, Humphrey claimed that we need to be able to customize and tailor the process to maximize individual enjoyment. Not

only is this exceedingly complicated, but the needs of the individual will change over time. Feiler noted that our objectives are different at different times in the process. For example, there might be fewer restrictions early in the process and more later.

The primary issue here, according to Balzer, is having an adaptable process that strikes a balance between prescriptiveness and proscriptiveness. An important question is whether the difficulty in adequately describing the process is a result of the formalisms that we have. On the one hand our formalisms are very weak; on the other hand, adequately describing (in advance) policy dynamics is exceedingly difficult.

Dowson reiterated his position that the fundamental issue is not whether the process is prescriptive or proscriptive but whether the human beings are part of the virtual machine on which the process descriptions run or whether the process description defines the behavior of the virtual machine on which the human process runs. The former implies more and more precise description while the latter provides effective support with minimal constraint. Osterweil responded that we need adequate formalisms for both prescription and proscription and the wisdom about when to use which. Moreover, pre and postconditions go far towards meeting this goal.

Balzer disagreed with Osterweil about pre and postconditions being very good. They do define some part of the process, but not all. What is left unstated is important as well — for example, what is not supposed to happen. This is referred to in AI as commonsense reasoning. Osterweil responded that he was exploring the tension between rules and procedures. If there are too many rules and not enough procedural content, problems will result. What we need is a steady, constant adjustment of the process. Balzer: still too weak.

The idea of modeling people in the process is seductive, noted Humphrey, but is vastly too complex. At most they should be partially in the process. Minsky brought up the problem of consensus: someone initiates an action, others agree or disagree. The actual process is not so simple to describe; it is more than just getting a yes or no answer. The solution is shaped by the contribution of each participant in reaching consensus.

There were other questions that Redwine thought were important. What sort of process is the setting up process and getting it established? How do you run it continuously and improve it? What about indoctrination and practice facilities — that is, how do you introduce the process to future participants?

Narayanaswami noted that teams are going to be a problem regardless of what process support is or is not provided. Important technical areas where formalisms are lacking (and their possibility of success) are 1) crowd control — extending transaction/object management (doable); 2) deliberation/negotiation/permissions/consensus — team process for change (worth looking at); 3) making the process state visible as ongoing justification for individual restrictions (doable, with difficulty in rule-based descriptions); and 4) goal decomposition — the interplay between individual and group goals (not doable).

The discussion was brought by Kishida-san back to the kinds of goals and objectives and are important in

the team process. Given that there are general social goals at the organizational level, such as success of the project, what are the conditions necessary to achieve personal goals at the individual level? Furthermore, what is the superimposing mechanism for these personal goals?

Ambriola claimed that process evolution is related to the human component. Moreover, process programs are derived from human behavior: we have human control that eventually leads to automation (in which there is no human then left). Perry objected that we will not automate ourselves completely out of the process and will thus have to support the non-automatable as well as the automatable.

Kelner proposed a third floor in the metaphorical house of software: a floor for the customer, sponsor or user that provides a peephole into the other floors. This peephole provides visibility into project team matters in terms of both product and process. This addition is consistent with Curtis' "layers" work and orthogonal to Perry's view. Derniame noted that there is communication between floors — for example, there are two process views, one for management and one for the developer; these two views can share messages, rules, operators, etc.

A slightly different definition of process was offered by Sugiyama: process is where people communicate — for example in UNIX we use local directories for sharing data and shell for local functions; thus, a description of the environment is a process program. Dowson asked if services change with time, do we have different environments or one that evolves. Sugiyama indicated that we might have both.

Thomas noted that one might choose an erroneous solution because of a misunderstanding of the rationale. Some models make these problems easier to deal with than others — for example, planning makes goals clear, most rule-based systems don't. Balzer noted that even with backtrace, the rationale is still an issue; rule-based systems are too flat. Huff reiterated Thomas's point: it is important to capture the state of the process by keeping the history. This is important for two reasons: first it is an interesting structure in its own right; second, it is the hierarchy of rules that helps get at the rationale. Heimbigner noted that we have a data representation that is an abstraction from the real system. We need that kind of representation and abstraction for rules.

Perry then redirected the discussion back to the topic of team effort drawing on aspects of his incremental iteration example that has dynamic product and policy-directed instantiations of processes for the individual developers cooperating in making changes to the product. What kinds of mechanisms did the various people use in the standard example to solve the interactions among the various roles? Dowson in his role as traffic cop for this session noted that this call for redirection back to the topic of team efforts was a discussion stopper. Apparently, it is still too early for this topic (as it was for the problems of scale at ISPW4).

Ambriola responded that defining roles is enough for the moment because we don't have them. It is enough of a problem to define an environment for a single person. [Perry's editorial comment: that is because the focus is on the human as part of the virtual machine, not on the behavior of the virtual machine on which the human process runs.] Penedo noted that a demonstration of a single user does not necessarily mean

anything about the multi-user version. They had run into deep issues and problems in going from a single-use to a multi-user workstation. Derniame supported the notion that the concept of role is very important and must be separated from that of the person operating in that role.

Two mechanisms used by Scacchi for team coordination were automatic mail and status information in documents. He sees the need for more high level policy descriptions and also the need for low level support modes for sharing. Cheatham indicated that he has support for multi-person process running and will be glad to run an example.

Kishida-san wrapped up the discussion by contrasting three viewpoints: the user's viewpoint is that of product-oriented project goals; the manager's viewpoint is that of the process model and supporting environment; the developer's viewpoint is process oriented and centered around individual goals. We have a mapping from user to developer. Why not have a mapping the other way?

Two types of ideal processes were then delineated: Confucianism and Taoism. In Confucianism, there is minor, peaceful integration with a detailed code of conduct and a law enforcement mechanism. These are represented by the dynasties after Emperor Wu of Han. Within this process is it very easy to be demoralized. In Taoism, there is great homogeneity with a minimal set of rules and a set of good examples of behavior. These are represented by a small number of minister in ancient dynasties before Han. The problem is that this process is difficult to implement. Which one is software engineering? Perry quipped that software engineering is "confusionist".