A Collaborative Spiral Software Process Model Based on Theory W

Barry Boehm and Prasanta Bose
USC Center for Software Engineering
University of Southern California
{boehm,bose}@sunset.usc.edu

1.0 Overview

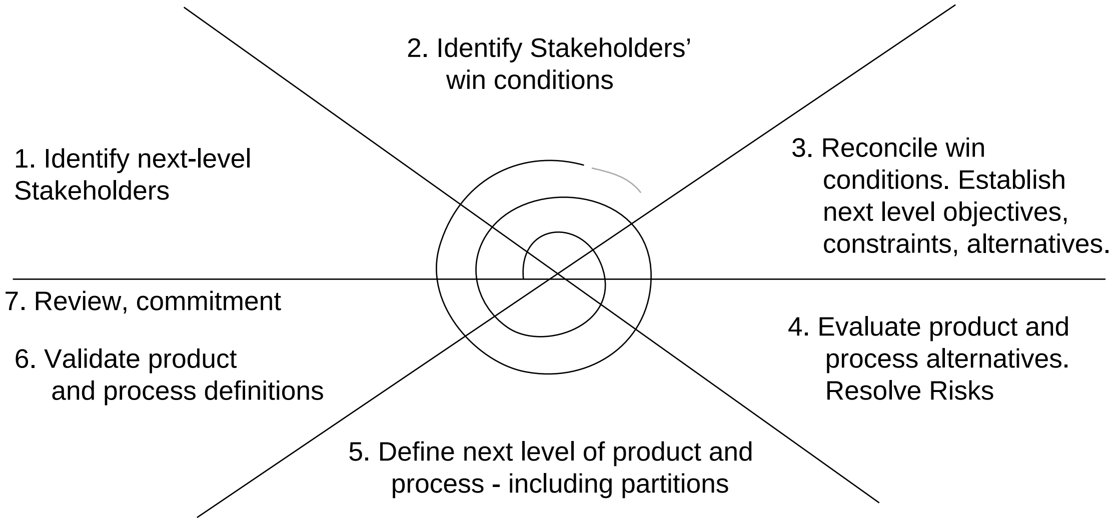
The Spiral Model of software development [Boehm, 1988] begins each cycle of the spiral by performing the next level of elaboration of the prospective system's objectives, constraints, and alternatives. A primary difficulty in applying the spiral model has been the lack of explicit process guidance in determining these objectives, constraints, and alternatives. This paper presents an extension of the spiral model, called the Next Generation Process Model (NGPM), which uses the Theory W (win-win) approach [Boehm-Ross, 1989] to converge on a system's next-level objectives, constraints, and alternatives. This Theory W approach involves identifying the system's stakeholders and their win conditions, and using negotiation processes to determine a mutually satisfactory set of objectives, constraints, and alternatives among the stakeholders.

This paper provides a summary of the resulting NGPM. It reports on experiences in applying it to a large Department of Defense program; in developing and experimenting with an initial support system for it; and in developing a refined model and support system based on the results of the experiment. The paper also compares the NGPM approach with related work in collaborative software processes and computer supported cooperative work (CSCW), and presents a set of conclusions from the work to date.

2.0 Key NGPM Features

Figure 1 illustrates the Theory W extensions to the Spiral Model that form the conceptual basis for the NGPM. The two additional sectors in each spiral cycle, “Identify Next-Level Stakeholders” and “Identify Stakeholders' Win Conditions,” and the “Reconcile Win Conditions” portion of the third sector, provide the collaborative foundation for the model. They also fill a missing portion of the current Spiral Model: the means to answer, “Where do the next-level objectives and constraints come from, and how do you know they're the right ones?” The refined Spiral Model also explicitly addresses the need for concurrent analysis, risk resolution, definition, and elaboration of both the software product and the software process. In particular, the nine-step Theory W process translates into the following Spiral Model extensions:

FIGURE 1. The NGPM Theory W extensions to the Spiral Model



- Determine Objectives. Identify the system life-cycle stakeholders and their win conditions. Establish initial system boundaries, external interfaces.
- Determine Constraints. Determine the conditions under which the system would produce win-lose or lose-lose outcomes for some stakeholders.
- Identify and Evaluate Alternatives. Solicit suggestions from stakeholders. Evaluate them with respect to stakeholders' win conditions. Synthesize and negotiate candidate win-win alternatives. Analyze, assess, and resolve win-lose or lose-lose risks.
- Record Commitments, and areas to be left flexible, in the project's design record and life cycle plans.
- Cycle Through the Spiral. Elaborate win conditions, screen alternatives, resolve risks, accumulate appropriate commitments, and develop and execute downstream plans.

A project following the NGPM will involve its stakeholders (users, customers, developers, maintainers, interfacers, software product line managers, ombudsmen, others) in a good deal of collaboration within the NGPM structure. To explore the parameters of effective software engineering environment (SEE) support of such collaboration, we worked out concepts for a Next Generation Process Support System (NGPSS), and have developed two successive prototypes of it, called NGPSS-0 and NGPSS-1. Experiences and lessons learned are discussed in Section 4.

3.0 A Large Experimental NGPM Application: The STARS Program

The U.S. DoD (Department of Defense) STARS (Software Technology for Adaptable, Reliable Systems) Program began in 1982 as an integrated program to address the overall ensemble of DoD software problems. By 1989, it was focused on contracts with three prime contractors (Boeing, IBM, and Unisys) and their subcontractor teams, to develop a set of prototype software engineering environments (SEE's) for DoD use. However, there were major mismatches between the program's planned products and the needs of its prospective government and industry users, operators, and maintainers. These shortfalls were in such areas as tool support, tool integration, tailorability, robustness, compatibility with CASE tools, portability, and maintenance costs, which were expected to be borne by DoD.

One of the authors (Boehm) assumed responsibility for the STARS program on his arrival as a Defense Advanced Research Projects Agency (DARPA; now ARPA) office manager in late 1989. He and the program's prospective new program manager, Dr. Jack Kramer, prepared to apply the spiral model to address the program's risks. They found that a serious set of risks involved incompatibilities among the expectations of the program's stakeholders. They decided to enhance the spiral model with a Theory W approach to determine whether a win-win solution for STARS was feasible (and if not, to discontinue the program).

As indicated in Section 2, the first two steps in the NGPM are to identify the system's stakeholders and their associated win conditions. Table 1 summarizes the results of these steps for STARS.

As generally happens, the union of the stakeholders' win conditions produced an overconstrained situation. The STARS prime contractors were government contracting companies or divisions, and were not prepared to commercially sell and service the STARS SEE's. But without commercially supported SEE's, DoD could not afford to operate and maintain them. Thus, for the program to remain viable, it became necessary for the STARS prime contractors to find commercial counterparts willing to sell and service the STARS SEE's. Eventually, each was able to do so: Boeing with DEC, IBM Federal Systems with IBM Canada, and Unisys with Hewlett-Packard.

However, although the commercial counterparts were willing to develop SEE's supporting software development in the DoD-mandated Ada programming language, they were not willing to use Ada for programming their own SEE software, as had been the previous STARS requirement. Their rationale was that their existing investments in C software, and their need to support C for commercial SEE customers, made it a much more cost-effective solution to program in C. Given that such a cost-benefit rationale fit DoD's Ada waiver criteria, ARPA was able to create a win-win solution by waiving the Ada programming requirement for the STARS SEE's.

Similarly, a number of other overconstrained situations were resolved into win-win situations for the stakeholders in Table 1. Some additional resulting features of the revised STARS program were [Bamberger, 1990]:

- Reorientation around much stronger software process and reuse support, to achieve software productivity and quality win conditions.
- Inclusion of a set of three demonstration projects, jointly sponsored by ARPA and a DoD Service (Army, Navy, Air Force), to reduce the risks of subsequent STARS SEE adoption.
- Negotiation of a set of common open STARS SEE interface specifications, to enable CASE vendors to reach a larger marketplace and reduce tool re-hosting costs.
- Addition of several STARS affiliates' programs, to provide CASE vendors, DoD Service organizations, and other DoD software contractors with access to intermediate STARS products and a voice in the STARS evolution strategy.

TABLE 1. STARS stakeholder win conditions

Stakeholder Class	Win Conditions
STARS prime contractors and their commercial counterparts	• Software Engineering Environment (SEE) sales • DOD acceptance of commercial SEE product line • Productivity leverage on primes' software business • Satisfied customers and users
STARS sub-contractors and CASE tool vendors	• Profits from large tools marketplace • Reduced tool re-hosting costs • Open architecture, multi-platform, polylingual • Stable evolution, voice in evolution strategy
Other DoD software contractors	• Productivity leverage on software business • Open architecture, multi-platform, ease of extension • Rapid availability, ease of use, reasonable cost • Stable evolution, voice in evolution strategy
DoD software support organizations	• Life-cycle effectiveness • Similar concerns to DoD software contractors • Support of software reengineering, Ada transition
DoD services and agencies	• Accelerator for, compatibility with Service/Agency software initiatives • Significant improvement in software productivity and quality • Low risks of SEE adoption on critical projects
ARPA, Congress, taxpayers	• All of win conditions above • SEE life-cycle affordability

Currently, STARS is successfully tracking its budget, schedule, and product plans. Under its current program manager, John Foreman, the demonstration projects are well underway, with significant buy-in by the various stakeholders.

However, one of the difficulties in applying the Theory W extension of the spiral model to STARS involved the frequent need to get representative, empowered stakeholders to the same place at the same time to negotiate win conditions. It took 5 months to go from the initial proposed STARS redirection to a set of prime contractor success plans embodying the resulting win-win approach. One clear opportunity to reduce this cycle time is to provide automated support for the process. This would also enable better recordkeeping and the ability to better analyze the process to determine improvements. This is the topic of the next section.

4.0 Experimentation with Automated Support

We are currently evolving NGPSS-1, the second cut at an automated groupware support system for the NGPM. Our approach to defining NGPSS-1 is based on the perspective that NGPSS is both a complex information processing system and a prospective tool for defining appropriate objectives, constraints, and alternatives for complex information systems. Given this, we decided to test our initial NGPSS-0 system in a bootstrap experiment to determine its ability to help us define win conditions and negotiate win-win objectives, constraints and alternatives for NGPSS-1. Section 4.1 and 4.2 provides a brief description of NGPSS-0 and the experiment conducted to test specific NGPM/SS hypotheses. Section 4.3 describes the main limitations of NGPSS-0 which got expressed as a set of win conditions and consequently defined the objectives for NGPSS-1. The section describes specific features of NGPSS-1 that met the objectives and hence addressed the limitations of NGPSS-0.

4.1 NGPSS-0 Overview

NGPSS-0 was implemented on top of Perceptronics’ CACE/PM®, a computer-aided concurrent engineering system originally developed for multichip module design. CACE-PM provided facilities for defining, simulating and sequencing the enaction of software processes, using a modified Petri-net formalism [Madni, 1988]. CACE-PM also supported the definition and use of object oriented schema hierarchies. NGPSS-0 uses these to identify stakeholders and to capture their win conditions; to represent conflicts, risks, and uncertainties (CRU’s) and to track candidate and adopted points of agreement (POA’s) on stakeholders’ negotiated win conditions. However, as indicated below, the experiment found limitations in the NGPSS-0 and CACE-PM schema support.

4.2 NGPSS-0 Bootstrap Experiment

We structured the experiment to be as representative as possible of the potential future uses of NGPSS. The experiment involved four representative participants playing the roles of NGPSS user, customer, developer and system engineer (whose role in NGPSS involves analyzing win conditions and sorting out candidate conflicts among them). The user role player had roughly 20 years’ experience in using and developing software tools. The customer role player had been a customer for many large software systems. The system engineer role player had considerable industrial systems engineering experience and served as system engineer in supporting NGPSS-0 during the experiment. The developer role player was one of the NGPSS-0 developers.

The experiment was also representative in that it ran for 6 weeks and was interpersed with the participant’s other activities. The four primary hypotheses tested during the experiment were:

1. *NGPSS-0 provides an adequate framework for supporting the NGPM*. One test of this hypothesis was the relative number and importance of the win conditions identifying deficiencies in NGPSS-0.

2. *The USC NGPSS win conditions will be comparable to a previous set of TRW software engineering environment (SEE) win conditions, gathered during the definition of the TRW Quantum Leap SEE in 1986-87*. This was determined by analysis of the TRW win conditions and entry of the most appropriate ones as NGPSS win conditions.

3. *Adding Theory W to the Spiral Model will generate the product and process objectives, constraints, and alternatives needed to initiate the original Spiral Model.* One test of this hypothesis was the relative number and importance of product and process objectives, constraints, and alternatives found among the win conditions entered.

4. *The bootstrap experiment will sufficiently define the most important top-level characteristics of NGPSS.* This is in the context that the bootstrap experiment covered only the initial cycle of an NGPSS spiral, focused on the two initial activities of win condition acquisition and identification of POA's and CRU's.

A detailed description of the experiment and its results is provided in [Boehm et. al.,1993a]. In summary, hypothesis 3 was basically confirmed in that 55 NGPSS win conditions included a good distribution of product and process objectives, constraints and alternatives. With respect to hypothesis 2, there was considerable commonality between NGPSS and TRW SEE win conditions, but also non-commonalities due to contextual differences in the two sets of organizational objectives and constraints.

With respect to hypothesis 1 and 4, the experiment identified several significant NGPSS-0 deficiencies, which have led to important new top level characteristics of NGPSS-1

1. Shared domain ontology;
2. Negotiation support
3. Low entry barrier for users;
4. Scalability;
5. Situated process definition;
6. Navigation support
7. Tailorability.

These are discussed next.

4.3 NGPSS-0 Deficiencies and NGPSS-1 Design Responses

4.3.1 Shared Domain Ontology

A frequent problem encountered in using NGPSS-0 was that the win conditions defined by two or more stakeholders were considered conflicting because of term mismatches and absence of information on the relation between terms. For example, in the experiment, two win conditions were identified by two different stakeholders: i) that tools should interoperate through the object base and ii) that tools should draw on knowledge base to ensure a consistent semantic data model of the data manipulated and communicated. Delays and confusion were encountered in determining whether these two win conditions were considered potentially conflicting because of term mismatches and lack of common model to understand their relationship. Moreover in NGPSS-0, it was very difficult to prioritize ones own win conditions since there was no model that made explicit the relations between each win condition.

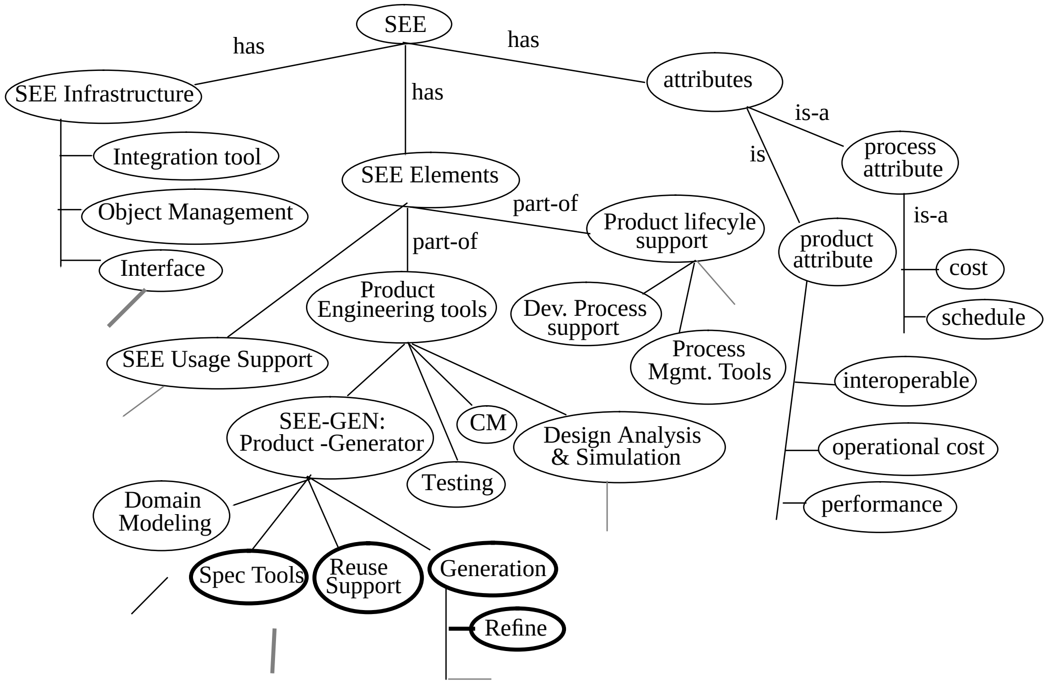
Both these problems stem from lack of a common domain model which provides a consistent semantic framework for all win conditions. Such a model acts to facilitate understanding of one's own and others’ win conditions and also the relation between the win conditions. In addition, to determine relations between two or more win conditions, a prerequisite to determine plausible conflicts, requires structuring them relative to the domain model. Failure to provide any domain ontology that can support the structuring results in differences in assumptions about the domain and consequent hindering the collaboration process as well as overburdening of each stakeholder. The latter leads to problems of scalability of the system.

Figure 2 shows part of the NGPSS-1 software engineering environment (SEE) domain taxonomy. The overall domain model has two components: i) a terminology database that provides a description of all key terms that get used in defining a win condition, and ii) a structured hierarchy of objects in the domain over which win conditions are expressed.

Viewing win conditions as desirable functional and non-functional constraints over the domain entities, the domain model reifies every entity in the domain as a first class object and defines a set of primitive relations in terms of “is-a”, “has-component” and “has-attribute”.

Had this taxonomy been available in NGPSS-0, the two potentially conflicting win conditions would have been entered in the same way: as a tool interoperability product attribute constraint, being implemented via the object management portion of the SEE infrastructure. If there indeed had turned out to be a substantive difference between the terms “object base” and “knowledge base”, the NGPSS-1 interactive editing capability for domain taxonomy and terminology could have been used to collaborate on a set of agreed-upon domain definitions and relationships.

FIGURE 2. Portion of NGPSS-1 SEE domain taxonomy



4.3.2 Negotiation Support

The second major problem encountered in using NGPSS-0 was handling conflicts arising from win conditions entered by different stakeholders. The experiment involved collaboration among multiple stakeholders. In NGPSS-0, the negotiation support was only in the form of capturing the end results of a negotiation activity. This severely limited any use of the system in supporting the negotiation for more complex conflicts. As a result, win condition clarification and resolution activities tended to involve more free-ranging interpersonal discussions, often using a physical whiteboard to identify and discuss options. Also there were no trade-off analysis tools to support negotiating amongst stakeholders by exploring trade-offs among stakeholders’ win conditions.

In analyzing how NGPSS could provide better support for resolving conflict via negotiation, we identified two major subproblems that the system must address: i) How to focus negotiation? and, ii) How to explore negotiated solutions? The first subproblem is addressed in NGPSS-1 by making use of explicit representation for option summaries. As illustrated in Figure 3, the option summaries capture general conflict resolution principles within a domain, which get specialized to the conflict context by the system engineer. For example, the principle of "Defer functionality" to reduce cost to avoid a cost conflict, does not commit to a specific functionality. The negotiation process context does the specific binding via exploring refinements of such an option through the use of tools. This brings us to the NGPSS-1 answer for the second subproblem of appropriate tools for exploring options and negotiated solutions. The current solution provided in NGPSS-1 is limited to an interactive Constructive Cost Model (COCOMO) tool for cost/schedule/functionality/performance trade-off analysis. Planned NGPSS enhancements in this area involve similar interactive tools for performance analysis and risk assessment.

FIGURE 3. An example of NGPSS-1 conflict and option summary

4.3.3 Low Barrier Entry Conditions

The CACE-PM support framework was graciously provided free of charge to USC by Perceptronics. However, the use of NGPSS-0 by any of the USC Center for Software Engineering's government and industry affiliates would have required them to pay a significant license fee for CACE-PM. This constituted a high entry barrier for non-USC users of NGPSS-0. One of the win conditions entered in the experiment by the NGPSS customer (also representing the Affiliates as stakeholders) was to have a low entry barrier for Affiliate users in future versions of NGPSS.

The resulting NGPSS-1 implementation decision was to restrict NGPSS-1 infrastructure packages to commodity software packages. Reflecting other user and customer win conditions on NGPSS portability, stability, and robustness, the commodity software packages chosen (X-Windows, Motif and Tool Talk) were also mature and based on open interfaces. As a result, there has been considerable interest by Affiliates in importing NGPSS-1 for experimental use.

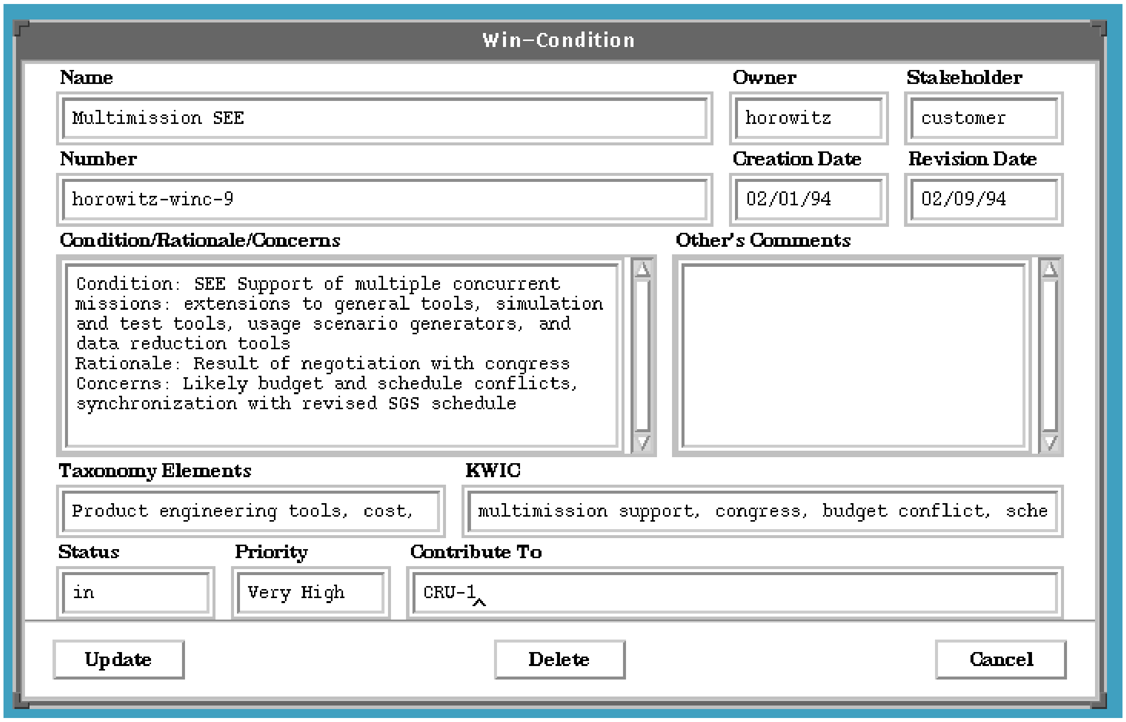
4.3.4 Scalability

Scalability of NGPSS-0 was strongly biased towards process definition. The CACE-PM schemas for activity elements allowed definition of process models at different levels of abstraction - where each level is a decomposition of the higher one. In NGPM, the process actions are situated and do not have hierarchical levels of decomposition. The problem of scalability arises from the complexity of managing the artifacts - win conditions, conflicts, and points of agreement - that result from stakeholder actions as opposed to planning and execution of complex process instances. This resulted in NGPSS-0 not being able to scale up with the complexity of win conditions, conflicts, and points of agreement for complex and large systems.

In NGPSS-1, the scalability issue is handled by i) explicitly distinguishing the representations for win conditions, conflicts and points of agreement, and ii) providing a level of structuring that attempts to strike a balance between the tasks of the stakeholder and the support tool. The specialized representation makes semantic distinctions based on the notions of win condition, conflicts and points of agreement, that can be used by the support system for the tractable management of the artifacts that result from the process actions.

An example of the win condition schema instance for multimission SEE for satellite ground stations is shown in Figure 4. The attributes - description, rationale and concerns specify respectively the objective, the argument for the objective, and concerns for risks, conflicts and uncertainties, relative to the individual stakeholders perspective. The domain elements attribute makes explicit the conjunction of taxonomy elements over which the objective defines desirable constraints. The status and contribute-to attribute makes explicit whether the win condition is part of the active context and its relation to points of agreements and conflicts. Similar schema descriptions are defined for points of agreement and conflicts. We will not go into the detail of these schemas but it is worth noting that the conflict schema (Figure 3) make explicit the elements of the context of win conditions and POAs which are in conflict and the set of possible options for resolving the conflict via negotiation.

FIGURE 4. An example of the win condition schema instance



Name: Multimission SEE
Owner: horowitz
Stakeholder: customer
Number: horowitz-winc-9
Creation Date: 02/01/94
Revision Date: 02/09/94
Condition/Rationale/Concerns: Condition: SEE Support of multiple concurrent missions: extensions to general tools, simulation and test tools, usage scenario generators, and data reduction tools
Rationale: Result of negotiation with congress
Concerns: Likely budget and schedule conflicts, synchronization with revised SGS schedule
Other's Comments:
Taxonomy Elements: Product engineering tools, cost,
KWIC: multimission support, congress, budget conflict, sche
Status: in
Priority: Very High
Contribute To: CRU-1

4.3.5 Situated Process Definition

NGPSS-0 was organized around prespecified processes which also defined the sequencing of the activities. The assumption made in NGPSS-0 was that the process would be generated first and then executed. In a collaborative context, most actions are primarily driven by the changes in states of artifacts resulting from other stakeholder's actions. This dynamic determination of the process actions based on the situation posed a limitation on the flexibility of the NGPSS-0 in allowing stakeholder's to enact activities relevant to the context of win conditions or point of agreements or conflicts.

In a collaborative software process - activities are determined not only by specific process goals but also heavily determined by artifacts and their states resulting from actions of other individuals. The latter provides a product oriented and situation specific bias on the process elements. NGPSS-1 facilitates capturing such a bias in two ways: i) Implicit use of product oriented process environment - process activities in NGPSS-1 are product or artifact centered, and ii) Tools to support determination of relevant activities - this is done through use of trigger mechanisms which notify changes in the object base and make recommendations for relevant actions.

4.3.6 Navigation Support

A basic capability needed to support collaborative process is querying the state of the artifacts resulting from such a process. That is, for a stakeholder to make an informed decision at a specific choice point, he must be able to view his and other's win condition/ POAs and CRU and their relations to other elements in the object base. Such a capability requires explicit representation of the dependencies between those elements that facilitate tracing of POAs to win conditions and support for query handling. NGPSS-0 lacked any capability to provide any such support; the need for such support was expressed in various ways in several of the experiment stakeholders' win conditions.

In the NGPM, the ontology of artifacts or products constitute three types of entities i) win conditions ii) points of agreements and iii) conflicts, risks and uncertainties. NGPSS-1 maintains the database of instances of these schemas and provides the capability to navigate through the database through query by example. So, for example, the customer can find all the high priority user specific win conditions that are active and relevant to product engineering tools by partially instantiating a win conditon template with the values provided for only the relevant attributes of stakeholder, status and taxonomy-elements.

4.3.7 Tailorability

During the NGPSS-0 experiment, a number of deficiencies were found in the win condition, CRU, and POA schemas. They needed capabilities for expressing such items as priorities, relations, comments by other stakeholders, and status. In principle, the CACE-PM schema support capabilities permitted these to be tailored as we learned more about what was needed. But in practice, our NGPSS-0 implementation had some inflexibilities which made tailoring difficult. The need for tailoring was entered by three of the stakeholders, and agreed to by the fourth.

As a result, NGPSS-1 was designed to make certain classes of tailoring very easy and other classes reasonably easy. Modifying domain terminology and taxonomies can be done interactively, as can adding options and relationships. Modifying win condition, POA, CRU, and option summary schemas is considerably easier, although the fundamental issues of schema migration and linking new schema elements to operations still need to be addressed.

5.0 Summary and Related Work

5.1 Related Work

Collaboration support often involves two dimensions of technical support. One dimension involves the information structure for coordination and communication. In this sense, the NGPSS approach to coordination support falls into the category of structured language-action systems exemplified by Flores and Winograd's Coordinator [Flores et al., 1988]. The initial NGPSS approach is aimed at providing more structure than the Coordinator, but not as much as attempted in gIBIS [Conklin-Begeman, 1988], SIBYL [Lee, 1990], and REMAP [Ramesh-Dhar, 1992], which have difficulties in scaling up to large systems. The NGPSS degree of automation is roughly similar to that of Object Lens [Lai et al., 1988] which brought attention to task coordination through rule-based processing of structured e-mail messages; and ISHYS [Garg-Scacchi, 1989] which tied rule-based hypertext and message management to software process support. NGPM and NGPSS are also trying to provide stronger support for scalable shared ontologies and for collaboration objectives via its experience-based domain-oriented prompts, defaults, and suggestions; and on the conceptual bases for collaboration and software/system development provided by Theory W and the Spiral Model. For example, the objective of achieving a win-win solution among stakeholders' win condition provides a much more explicit answer to the question, "What are we trying to collaborate about?", than most other conceptual frameworks for collaboration.

Another dimension of collaboration support involves technical support for information exchange, sharing, and visualization [ACM, 1993]. The NGPSS approach has initially focused on getting a solid concept of operation and information framework established for the system. Subsequently we plan to explore incorporation of more collaboration support tools including hyper-media communication as exemplified by the "Messages" program in the Andrew Message System [Borenstein et al., 1991], and computer-aided meetings represented by WYSIWIS [Stefik, 1987].

Considering the use of NGPM/SS to the task of requirements engineering, our work is related to the work of [Nuseibeh et. al., 1993; Easterbrook 1991; Dardenne 1991; Feather 1989] on requirements engineering for composite systems based on use of multiple viewpoints. Nuseibeh uses the representation of viewpoint to locally capture partial product representation knowledge, process knowledge, specification knowledge and its domain. The win condition schema used in NGPSS, can be viewed as a higher level abstraction of a viewpoint template: the abstractions postpone decisions on more precise product representation aspects to later stages of the NGPM and allows semi-precise representations of functions, cost, performance and schedule that are relevant to the process of collaboration for initial requirements engineering. Also, in NGPSS the process actions being primarily situated communication and negotiation/coordination actions, are left implicit in the NGPSS environment to allow dynamic determination of the process actions.

In another vein, the interactive approach to conflict exploration and resolution in Synoptic support system [Easterbrook 1991] has similarities to interactive searching of win-win arrangements leading to generation of POA's and CRUs. One of the limitations of Synoptic that NGPSS tries to address is multi-party involvement in the process: Synoptic allows only two viewpoints to be compared at once. One of the major current NGPM/SS efforts involve translation of stakeholder consensus Points of Agreement into build-to specifications capable of supporting multi-stakeholder views.

NGPM/SS also has similar objectives to JAD and the European Participatory Design Approaches [Carmel et al., 1993]. Its major differences arise from the use of Theory W as a basis for collaborative requirements generation, use of domain specific representations and default domain specific knowledge to guide win condition elicitation and focus exploration of win-win solutions via POA's and CRUs. The NGPM/SS approach also involves the use of performance and cost trade-off analysis tool such as COCOMO, that can be used as tools for evaluating win conditions and determining CRUs, and negotiating win-win solutions. In this respect, NGPSS has commonalities with the CSCWE work

[Palmer et al, 1992] on use of decision support tools to facilitate consensus building in collaborative requirements engineering.

5.2 Summary

In this paper we described a collaborative spiral process model based on theory W, described some of the key features of an semi automated process support system for such a model. The key insight underlying the process model is based on the understanding that a software product has a life cycle which engages multiple stakeholders at different stages. The stakeholders impose heterogenous constraints called win conditions on the product. Ignoring such constraints results in unsuccessful products and projects associated with them. The solution provided by NGPM is based on concurrent engineering of the software product and process lifecycle that attempts to define objectives, constraints and alternatives on a system through a collaborative process. It uses Theory W to manage individual stakeholder concerns and search for win-win solutions. Results of large-scale non-automated use of the Theory W/Spiral Model approach, and of small-scale experimentation with automated support for the approach, indicate that the approach is scalable and beneficial.

6.0 References

[ACM, 1993]. Communications of the ACM, “Multimedia in the Workplace”, CACM, Vol. 36, Number 1, January 1993.

[Bamberger, 1990]. J. Bamberger, ed., “STARS/Users Workshop: Final Report - Issues for Discussion Groups,” CMU/SEI-90-TR-32, Software Engineering Institute, Pittsburgh, PA, December 1990.

[Boehm, 1988]. B.W. Boehm, “A Spiral Model of Software Development and Enhancement,” Computer, May 1988, pp. 61-72.

[Boehm-Belz, 1988]. B.W. Boehm and F.C. Belz, “Applying Process Programming to the Spiral Model: Lessons Learned,” Proceedings, IEEE 4th Software Process Workshop, May 1988.

[Boehm-Ross, 1989]. B.W. Boehm and R. Ross, “Theory W Software Project Management: Principles and Examples,” IEEE Trans. Software Engr., July 1989.

[Boehm et. al., 1993]. B. W. Boehm, P. Bose, E. Horowitz, W. Scacchi, S. Bendifallah and A. Madni, “Next-Generation Software Processes and Their Environment Support”, Proceedings, USC Center for Software Engineering Inaugural Convocation, June 1993.

[Boehm et. al., 1994. B. W. Boehm, P. Bose, E. Horowitz, and M-J. Lee, “Software Requirements As Negotiated Win Conditions”, Proceedings, 1994 International Conference on Requirements Engineering, IEEE, April 1994.

[Bandinelli et al, 1993]. S. Bandinelli, A. Fuggetta, and C. Ghezzi, “Software Process Model Evolution in the SPADE Environment”, to appear on TSE Special Issues on Process Evolution.

[Borenstein and Thyberg, 1991]. N.S. Borenstein and C.A. Thyberg, “Power, Ease of Use, and Cooperative Work in a Practical Multimedia Message System,” Int'l J. Man-Machine Studies, Vol. 34, No. 2, Feb. 1991, pp. 229-259.

[Carmel et al., 1993]. E. Carmel, R.D. Whitaker, and J.F. George, “PD and Joint Application Design: a Transatlantic Comparison”, Comm. ACM, Jun. 1993, pp. 40-48.

[Conklin-Begeman, 1988]. J. Conklin and M. Begeman, “gIBIS: A Hypertext Tool for Exploratory Policy Discussion,” ACM Trans. Office Info. Systems, Oct. 1988, pp. 303-331.

[Dardenne et al., 1991]. A. Dardenne, S. Fickas and A. van Lamsweerde, “Goal-directed Concept Acquisition in Requirement Elicitation”, Proceedings, Sixth International Workshop on Software Specification and Design, Oct. 1991, pp. 14-21.

[Easterbrook, 1991]. S. Easterbrook, “Handling Conflict between Domain Descriptions with Computer-supported Negotiation”, Knowledge Acquisition, Vol. 3, 1991, pp 255-289.

[Feather, 1989]. M.S. Feather, “Constructing Specifications by Combing Parallel Elaborations”, IEEE Trans. on Software Engr, 15, 1989, pp 198-208.

[Flores et. al., 1988]. F. Flores, M. Graves, B. Hartfield, and T. Winograd, “Computer Systems and the Design of Organizational Interaction,” ACM Trans. Office Info. Systems, April 1988, pp. 153-172.

[Garg-Scacchi, 1989]. P.K. Garg and W. Scacchi, “ISHYS: Designing an Intelligent Software Hypertext System”, IEEE Expert, Vol. 4(3), Fall 1989, pp 52-62.

[Lai et al., 1988]. K-Y. Lai, T.W. Malone, and K-C Yu, “Object Lens: A “Spreadsheet” for Cooperative Work,” ACM Trans. Office Info. Systems, October 1988, pp. 332-353.

[Lee, 1990]. J. Lee, “SIBYL: A Qualitative Decision Management System,” in Artificial Intelligence at MIT: Expanding Frontiers, P. Winston and S. Shellard, eds., MIT Press, 1990, pp. 106-133.

[Madni, 1988]. A.M. Madni, “HUMANE: A Designer's Assistant for Modeling and Evaluating Function Allocation Options,” Proc. Ergonomics of Advanced Manufacturing and Automated Systems Conf, Louisville, KY(August 1988)

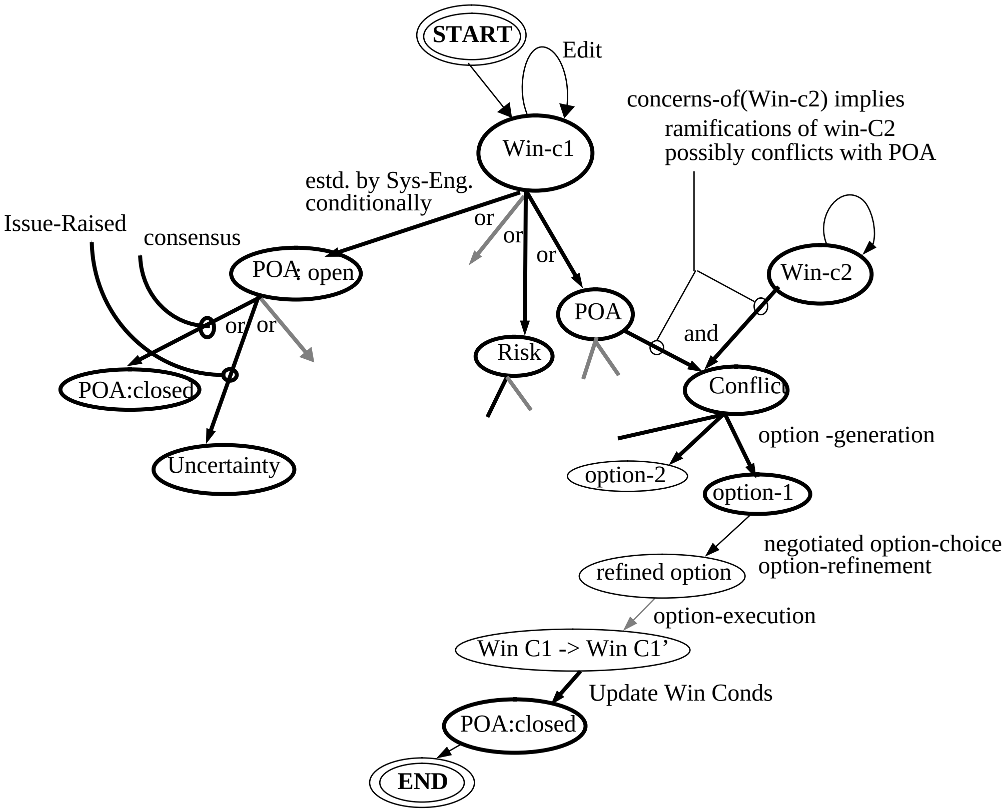
[Nunamaker et al., 1991]. J.F. Nunamaker, A.R. Dennis, J.S. Valacich, D.R. Vogel, and J.F. George, “Electronic Meeting Systems to Support Group Work,” Comm. ACM, Vol. 34, No. 7, July 1991, pp 40-61.

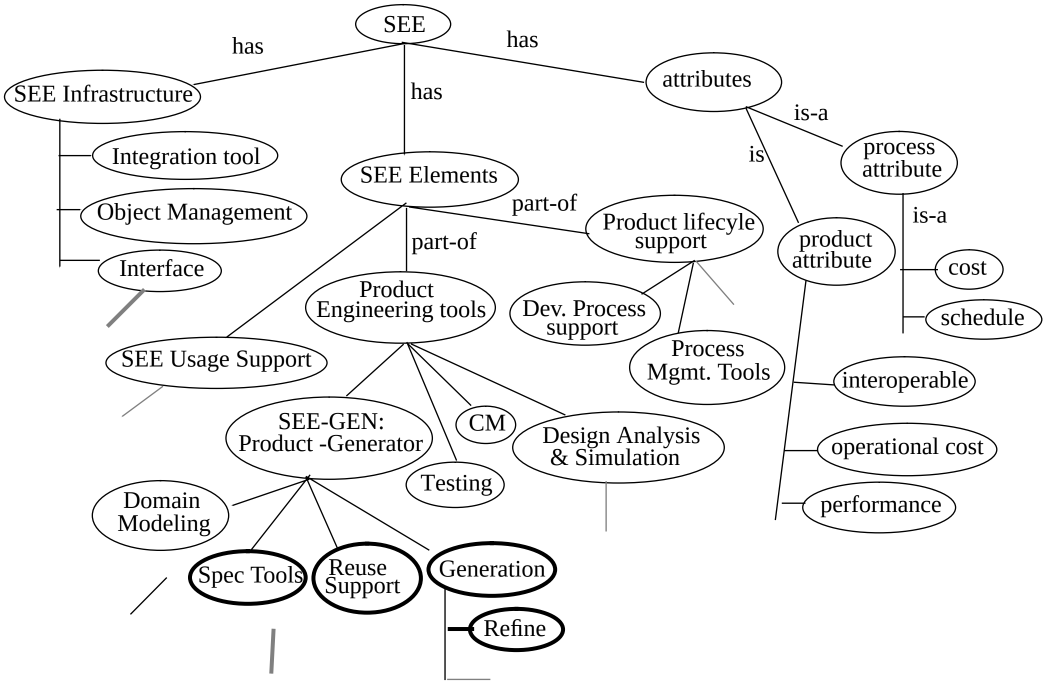
[Nuseibeh et. al., 1993]. B. Nuseibeh , J. Kramer, A. W. Finkelstein, “Expressing the Relationship Between Multiple Views in Requirements Specification”, Proceedings of the15th International Conference on Software Engineering, 1993.

[Palmer et al., 1992]. J. D. Palmer and N. Ann Fields, “An Integrated Environment for Requirements Engineering”, IEEE Software, Vol. 6, No. 3, May 1992.

[Ramesh-Dhar, 1992]. B. Ramesh and V. Dhar, “Supporting Systems Development by Capturing Deliberations During Requirements Engineering,” IEEE Trans. SW Engr., June 1992, pp. 498-510.

[Stefik et al., 1987]. M. Stefik, G. Foster, D.G. Bobrow, K. Kahn, S. Lanning, and L. Suchman, “Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings,” Comm. ACM, Vol. 30, No. 1, Jan. 1987, pp. 32-47.


START
Edit
Win-c1
concerns-of(Win-c2) implies
ramifications of win-C2
possibly conflicts with POA
estd. by Sys-Eng.
conditionally
or
or
or
Issue-Raised
consensus
POA: open
or
or
POA:closed
Uncertainty
Risk
POA
and
Win-c2
Conflict
option -generation
option-2
option-1
negotiated option-choice
option-refinement
refined option
option-execution
Win C1 -> Win C1'
Update Win Conds
POA:closed
END


SEE
has
has
has
SEE Infrastructure
attributes
is-a
Integration tool
SEE Elements
is
process attribute
Object Management
part-of
Product lifecyle support
is-a
part-of
product attribute
Interface
cost
Product Engineering tools
Dev. Process support
schedule
SEE Usage Support
Process Mgmt. Tools
interoperable
SEE-GEN: Product -Generator
CM
Design Analysis & Simulation
operational cost
Testing
Domain Modeling
performance
Spec Tools
Reuse Support
Generation
Refine