The CBSP Approach For Bridging Requirements and Architecture Models

Paul Gruenbacher

University of Southern California
Center for Software Engineering
941 W. 37th Place, Los Angeles, CA 90089-0781
Email: gruenbac@sunset.usc.edu

Requirements and the system architecture emerge in an iterative process that involves heterogeneous stakeholders with conflicting goals, expectations, and objectives [1]. The transition from requirements to the architecture in a non-waterfall way is an important open research challenge.

Approaches bridging requirements and architecture models have to find the right balance between flexibility and formality to be beneficial for all stakeholders involved in requirements engineering and system architecting. We have been exploring ways to integrate WinWin negotiation models [1] and the C2-style architectural models [5] in the course of the EasyWinWin [3, 2] and SAAGE projects [5,6].

We have developed the CBSP (*C*omponent, *B*us, *S*ystem, *P*roperty) approach that provides an intermediate model between requirements and architectures [4, 6]. WinWin artifacts are assessed in a tool supported voting process for their relevance to the system architecture's components, connectors (i.e., buses), overall configuration (i.e., the system itself or a particular subsystem), and properties (e.g., reliability, performance, and cost):

- C-artifacts describe or involve a Component in an architecture.
- B-artifacts describe or imply a connector (Bus).
- S-artifacts describe System-wide features or features pertinent to multiple components and connectors.
- CP-artifacts describe or imply Component Properties.
- BP-artifacts describe or imply connector (Bus) Properties.
- SP-artifacts describe or imply System Properties should pertain to the entire architecture.

Table 1 shows WinWin artifacts used in a specific example that have been refined into various CBSP dimensions. These dimensions ensure that the architectural concerns expressed in the WinConditions are captured in the resulting CBSP artifacts. The artifacts are taken from a requirement, architecture, and design modeling exercise we have carried out for a complex logistics application (cargo router).

Table 1: CBSP-based refinement of stakeholder win conditions

Dimension	*WinWin artifact*	*Refined CBSP artifact*
C	W12 Allow customizable reports, generated on the fly.	W12_C Report generator component
B	W30 The system should have interfaces to related applications (vehicle management system, staff availability).	W30_B Connector to staff and vehicle management system.
S	W3 Capability to react to urgent cargo needs.	W3_S The system should deploy automatic agents to monitor and react to urgent cargo needs.
CP	W44 The client UI component should run on a palm-top or laptop device.	W44_CP The client UI component should be portable and efficient to run on palm-top as well as laptop devices.
BP	W42 Integration of third party components should be enabled without shutting down the system.	W42_BP Dynamic, robust connectors should be provided to enable "on the fly" component addition and removal.
SP	W6 Operators must be notified of subsystem failures within three seconds.	W6_SP The system should support real-time communication and awareness.

CBSP artifacts are used (1) to refine existing WinWin artifacts (e.g., stakeholder win conditions, options) and to (2) capture feedback from architectural modeling and simulation (e.g., architectural issues, options). CBSP therefore facilitates the synthesis (classification and refinement) of negotiation artifacts into architectural elements and enables feedback from architecture modeling and analysis (architectural tradeoff issues and options).

References

[1] Boehm B., Egyed A., Kwan J., Port D., Shah A., Madachy R., "Using the WinWin Spiral Model: A Case Study", IEEE Computer, 7:33–44, 1998.

[2] Boehm B., Gruenbacher P., "Supporting Collaborative Requirements Negotiation: The EasyWinWin Approach", International Conference on Virtual Worlds and Simulation, San Diego 2000.

[3] Gruenbacher P, Collaborative Requirements Negotiation with Easy WinWin, 2nd International Workshop on the Requirements Engineering Process, Greenwich London, September 2000 (to appear).

[4] Gruenbacher P., Egyed A., Medvidovic N., "Dimensions of Concerns in Requirements Negotiation and Architecture Modeling", Workshop on the Multi-Dimensional Separation of Concerns in Software Engineering at ICSE 2000.

[5] Medvidovic N., Rosenblum D.S., Taylor R.N., "A Language and Environment for Architecture-Based Software Development and Evolution", In: Proceedings of the 21st International Conference on Software Engineering (ICSE'99), pp. 44-53, Los Angeles, CA, May 16-22, 1999.

[6] Medvidovic N., Gruenbacher P., Egyed A. F., Boehm B.W., "Software Lifecycle Connectors: Bridging Models across the Lifecycle", Technical Report, USC-CSE, 2000.