

Purpose:

To use continuous models to establish and maintain alignment between business requirements and enterprise system solutions.

Applications:

- Enterprise Architecture
- Business Process Analysis
- Business Process Modeling
- Requirements Management and Program Governance

Results:

- Better alignment between system solutions and business requirements
- Increased speed and ability to support compressed timelines and agile software development
- Reduced software design documentation costs due to automated output
- Increase ability to communicate system requirements to system development organizations

The Problem: Discontinuities between Requirements and Design

The transition from business process requirements to system solution design can be done using three system views: user interactions, information ontology, and business rules. Even then, in order to design a complete solution, system solution elements and business rules outside the system boundary under design need to be incorporated. That is, the interactions (input/outputs, messages, etc.) provided by system elements across the system boundary are needed to completely specify the requirement. In order to capture system requirements, “use cases” are used to describe the required system behavior. The substantive part of use cases is the description of the interaction between the actor and the subject system. This description is generally represented in a Microsoft Word™ narrative and is often difficult to update as design iterations occur. Also, use case models do not include the interactions between actor and the subject system due to the inability of the UML modeling language to support it.

The Solution: Continuous, Dynamic Business Process Analysis and Modeling

Our methodology, however, creates an extension to the business process model to describe these interactions, and can therefore provide design continuity with other business processes. The business process model, which has the user experience as a main focus, is used to include these interactions, not the system architecture. In doing so, we develop a new model to expand upon the business process model: the use case interaction model.

The Use Case Interaction Model (UCIM) describes the sequence of user interactions with the system for a specific use case. It enables the required behavior of the system, as seen by the user and as integrated into the business process, to be conceptually modeled, simulated, managed, and documented. The information ontology is a necessary part of defining requirements because it identifies and describes the information exchanged at the system boundary. Two modeling levels of the ontology can coexist – the overall information model, and the model segment that is used to define deliverables between activities. Our method captures the information requirements, models them, and includes them directly in the business process model.

Tooling for Process Continuity

In order to achieve speed and standardization of quality design artifacts, it is necessary to develop a process toolset that supports the analysis and maintains continuity. Although tool chain engineering is in an early stage and various mechanisms are evolving, it is nevertheless important that design artifacts can not only be produced in the correct paper form, but can be suitably passed along the process tool chain to those tools used for system solution architecture. Moving these artifacts electronically ensures that there is no distortion between the business process and the system solution.

Our process methodology bridges the divide from business process to system solution architecture, using a production suite that:

- Incorporates standard languages for the exchange of BPMN between the business process tools and system solution design tools
- Allows build and export of a transformation from one tool to another
- Uses an extended metamodel to provide continuity to both these areas

Ability to Support Compressed Timelines and Agile Development

Our production tool suite allows requirements documentation to be generated, not edited by hand in word processing software. This allows design changes generated from short design cycles typically associated with agile software development to be reflected rapidly in the documentation and any interchange format.

For more information, please contact either Mr. Tom Bonina or Mr. Tony Mallia at 703-896-7681, or email at tbonina@edmondsci.com or amallia@edmondsci.com.