

Sharing the Vision = Designs that Get Built

Duane Degler, IPGems (ddegler@ipgems.com)

Lisa Battle, Lockheed Martin (lisa.battle@ssa.gov)

Darrell H. Taylor, Lockheed Martin (darrell.h.taylor@ssa.gov)

INTRODUCTION

For designs to be successfully implemented, it is necessary that the different stakeholders and members of a multidisciplinary project team reach a shared understanding of the problem and the solution. User-centered design (UCD) is our approach, but in large-scale projects we are increasingly finding that the technical team members are using object modeling techniques based on the Unified Modeling Language (UML). Our goal is to share our links between these two disciplines, by exploring similarities and differences in approaches, goals, and techniques. There are natural points where UCD deliverables should connect with UML deliverables, and vice versa, to help a project team create and communicate a shared vision. We have prepared a poster that explores our integration of these methods and deliverables, based on our experience with projects where both UCD and UML are used.

THE IMPORTANCE OF COMMUNICATION

Communicating analysis findings and design ideas to different stakeholders and team members (keeping in mind their differing needs and perspectives) is a challenge that most of us face daily. It is important for everyone on project teams to have a clear, thorough understanding of what the system is expected to do, how user needs are addressed, and that the overall business vision and goals are represented in design and development.

We have found that if we don't make an effort to integrate our approaches, one or more of the following inevitably happens:

- UCD insights and designs don't influence system development, so user needs are not effectively met
- Designers and user representatives are required to work in formats that are "foreign," which either don't effectively communicate user needs clearly or are very difficult and time-consuming to use
- There is an increase in misunderstandings or misinterpretation of requirements
- The users and the project team can end up with bad outcomes and disappointments

UCD and UML both strive to define real user requirements and use them as the basis for system development, and each has a plethora of documentation options and formats. UCD has a richer design vocabulary, but it is important for that to fit with development definitions. UML practitioners describe the language as "architecture-centric, use case driven." Similarly, UCD could be described as "user-centered, goal driven." Because the emphasis speaks to different information needs, these approaches have the potential to complement one another when used together—as long as there is clear communication.

TOOLS TO DOCUMENT ANALYSIS AND DESIGN

Some of the main areas where we have encountered miscommunication or confusion are described below. We consider the following five areas particularly important points of integration.

Goals and Vision

The UCD approach to analysis typically begins with a discussion of the business goals and strategic value of the proposed system. Specific methods may include brainstorming sessions with stakeholders to articulate both short-term and long-term needs, as well as visioning sessions to stimulate thinking. The business goals are documented and referred to throughout the design process to influence design decisions (some teams hang business goals on the wall of the design room). They are also used when defining usability goals and performance metrics for testing, so that the tests results can show whether or not the system achieves the goals.

UML reference books emphasize the importance of the vision, although the purpose has been described as “summarizes the high-level requirements” and scope. However, there is not as clear a standard approach, particularly for identifying explicit business and user goals that are referenced throughout design and testing. In practice, we’ve experienced both UCD and UML advocates producing (and then ignoring) the vision... so we emphasize with our teams: don’t ignore it, value it!

Personas and Actors

Because personas and actors both describe the users of a system, some analysts initially assume that they are the same. However, they actually serve very different, complementary purposes. In UCD, personas are composite sketches of representative users, with richly detailed characteristics including their job experience, attitudes, personal preferences, knowledge of a subject area, and goals. They are used throughout the design process to keep the design team focused on the variety found among real users and to ensure that the eventual design meets the needs of the intended audience. Their weakness is that they can focus on who the users are, rather than what they do in regards to the system. Actors in UML are typically more “generic” and use case-focused. They represent *all types* of users (even hardware and software) that will interact with a specific use case or set of use cases within the proposed system. They help designers focus on functional goals. There is no specific modeling of user goals and intentions in UML, and the actor cannot help provide context to the interaction designer.

Scenarios and Use Cases

Because scenarios and use cases both describe tasks, again some team members may initially assume that they are the same. UCD scenarios are narrative stories from a user’s perspective describing a situation, triggered by a realistic event, in which a user interacts with the system. They are rich in context. Scenarios are used in design sessions, walkthroughs, and usability tests to ensure that the system design effectively supports users in a wide range of real-life situations. However, rarely will scenarios be written to cover every situation and detailed functional action. A use case, on the other hand, describes a specific usage of the system by one or more actors, and is far more “modular” than a scenario. Use cases are used to analyze and design robust system functions and architecture.

UCD scenarios and UML use cases are actually quite different, but can be very effectively used in combination. A UCD scenario typically encompasses several use cases. For example, making an online purchase may involve selecting items to buy, identifying your personal and account details, validating payment, and arranging shipping. UCD scenarios can help to ensure that all of the use cases, and the use cases that best support the business, are identified and tested. UML use cases can help designers think about consistent and reusable interaction styles across the system. They may also anticipate exceptions or unusual circumstances that are not described in scenarios. UML practitioners use the word “scenarios” to describe examples that illustrate variations or exceptions on a use case. We try to have a use case *refer* to multiple scenarios, rather than *contain* the scenarios.

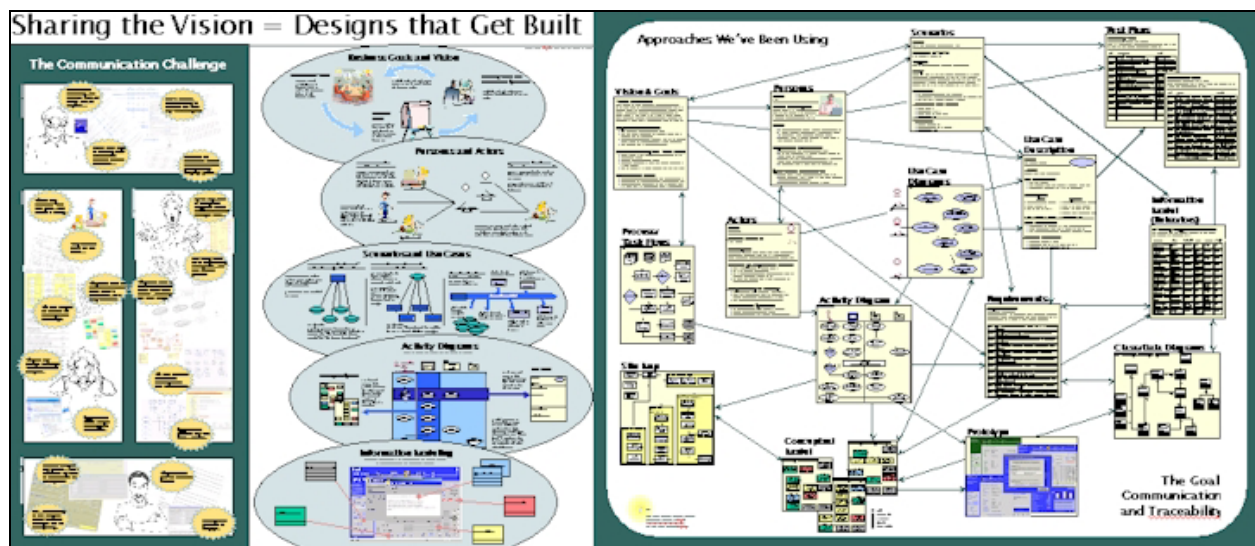
Diagrams, Site Maps and Task Flows

Various deliverables from UCD and UML illustrate the flow of activities, decision points, alternative branches, connections between system functions, and relationships with other systems or business components. These deliverables must be readable and understandable by the client/users in order to conduct walkthroughs and validate that the system will meet the goals. Activity diagrams and task flows model the way that the system supports the users' work. These flows then illustrate where and how particular functions (use cases) and views (site maps) relate within the system.

Information Models and Data Models

UCD addresses the design of everything that users see and interact with, including many types of information presented online. In many web applications, the design of the user interface is not just restricted to laying out the data – it also includes incorporating page titles, labels, help text, instructions, examples, policy references, knowledge resources, and other informational items that are not often represented in development specifications. System modelers may leave out information that is not transactional data. UCD encourages all types of information to be described and included as objects available for users. They need to be part of not only the UCD information model, but also the UML class diagrams/data model.

THE POSTER



FURTHER READING

1. Ambler, Scott. (2001). Be Realistic About the UML. <http://www.agilemodeling.com/essays/realisticUML.htm>
2. Booch, Grady, Rumbaugh, James, and Jacobson, Ivar. (1998) *The Unified Modeling Language User Guide*. Reading, MA: Addison Wesley.
3. Carroll, John M., Ed. (1995). *Scenario-Based Design*. New York, NY: John Wiley & Sons.
4. Constantine, Larry and Lockwood, Lucy. (1999). *Software for Use*. New York, NY: ACM Press.
5. Cooper, Alan. (1999). *The Inmates are Running the Asylum: Why High-Tech Products Drive us Crazy and How to Restore the Sanity*. Indianapolis, IN: SAMS.
6. Fowler, Martin. (1999). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Reading, MA: Addison Wesley.
7. Hudson, William. (2001). Toward Unified Models in User-Centered and Object-Oriented Design. In Mark Van Harmelen, Ed., *Object Modeling and User Interface Design: Designing Interactive Systems*. Reading, MA: Addison Wesley.
8. Kulak, Daryl and Guiney, Eamonn. (2000). *Use Cases: Requirements in Context*. ACM Press.
9. Rourke, Chris. (2002). Making UML the Lingua Franca of usable system design. *Interfaces Magazine* from the British HCI Group. http://www.uservision.co.uk/usability_articles/usability_uml.asp

ABOUT THE AUTHORS

Duane Degler has over fifteen years' consulting experience in user-centered design, information/knowledge strategy, process analysis, system implementation, and training. His recent activities include supporting the Usability Center at the Social Security Administration as a designer and strategist, working with a team from Lockheed Martin. Prior to that, his interface/system design work has received three US awards. He has managed pioneering multimedia projects in the '80s and spent much of the '90s in the UK involved in knowledge management research and consulting, contributing to his multi-disciplinary approach to interaction design. Duane holds a Masters degree in organizational communication.

Darrell Taylor has over twelve years' consulting experience designing user interfaces, supporting system development and creating performance-centered solutions. Working with Lockheed Martin, he is currently supporting the Social Security Administration Usability Center, responsible for leading UI design efforts for agency-wide Web-based applications, mentoring and educating project teams on user-centered methods, and consulting on business process analysis efforts. Mr. Taylor holds a Masters degree in computer systems management.

Lisa Battle has consulted on interface usability and led the design of software applications and web sites for a variety of government and commercial clients. In her current position at Lockheed Martin, she is responsible for user-centered design for web-based applications at the Social Security Administration and mentoring project teams in user-centered methods. She holds a Masters degree in cognitive psychology/human factors from George Mason University.