

**IUSED 2008**

Pia Stoll

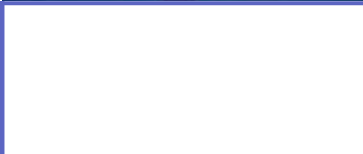
Industrial PhD student at Mälardalen  
University

Research Engineer at Industrial Software  
Systems, ABB Corporate Research

Len Bass, Bonnie E. John, Elsa Golden  
Human Computer Interaction Institute  
Carnegie Mellon University



# Preparing Usability Supporting Architectural Patterns for Industrial Use



# ABB Research Context

- Asea was founded 1890 and Brown Boveri founded 1891 merged 1988 to form ABB
- Today ABB has 120 000 employees worldwide
- The five ABB businesses;
  - Power Products
    - Power Products are the key components to transmit and distribute electricity
  - Power Systems
    - Systems and services for power transmission and distribution grids, and for power plants
  - Automation Products
    - Products to improve customers' productivity, including drives, motors and generators, low voltage products, instrumentation and analytical, and power electronics
  - Process Automation
    - Integrated solutions for control, plant optimization, and industry-specific application knowledge
  - Robotics
    - Industrial robots - also providing robot software, peripheral equipment, modular manufacturing cells and service for specific tasks such as welding, painting and finishing, picking, etc
- The Corporate Research program "Industrial Software Systems" assists the businesses in;
  - New platform projects
  - New technology projects
  - Re-factoring projects
  - Application projects
  - Assessment projects
- We team up with the businesses' software architect, management, developers, etc but we can also do pre-studies without the businesses taking active participation.

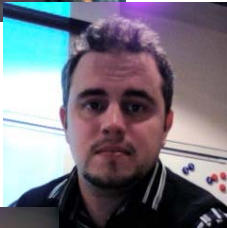


# Usability and Software Architecture

## ■ ABB Research team



- Pia Stoll
  - Research Engineer (Software Engineering)
  - ABB Corporate Research



- Fredrik Alfredsson
  - Research Engineer (Usability)
  - ABB Corporate Research



- Sara Lövmemark
  - Usability Engineer (Applications and Requirements)
  - ABB Robotics

# Usability and Software Architecture

- Carnegie Mellon University Research team



- Len Bass

- PhD, Computer Science, Purdue, 1970
- Faculty, University of Rhode Island, 1970-1986
- Software Engineering Institute, Carnegie Mellon, 1986-present



- Bonnie E. John

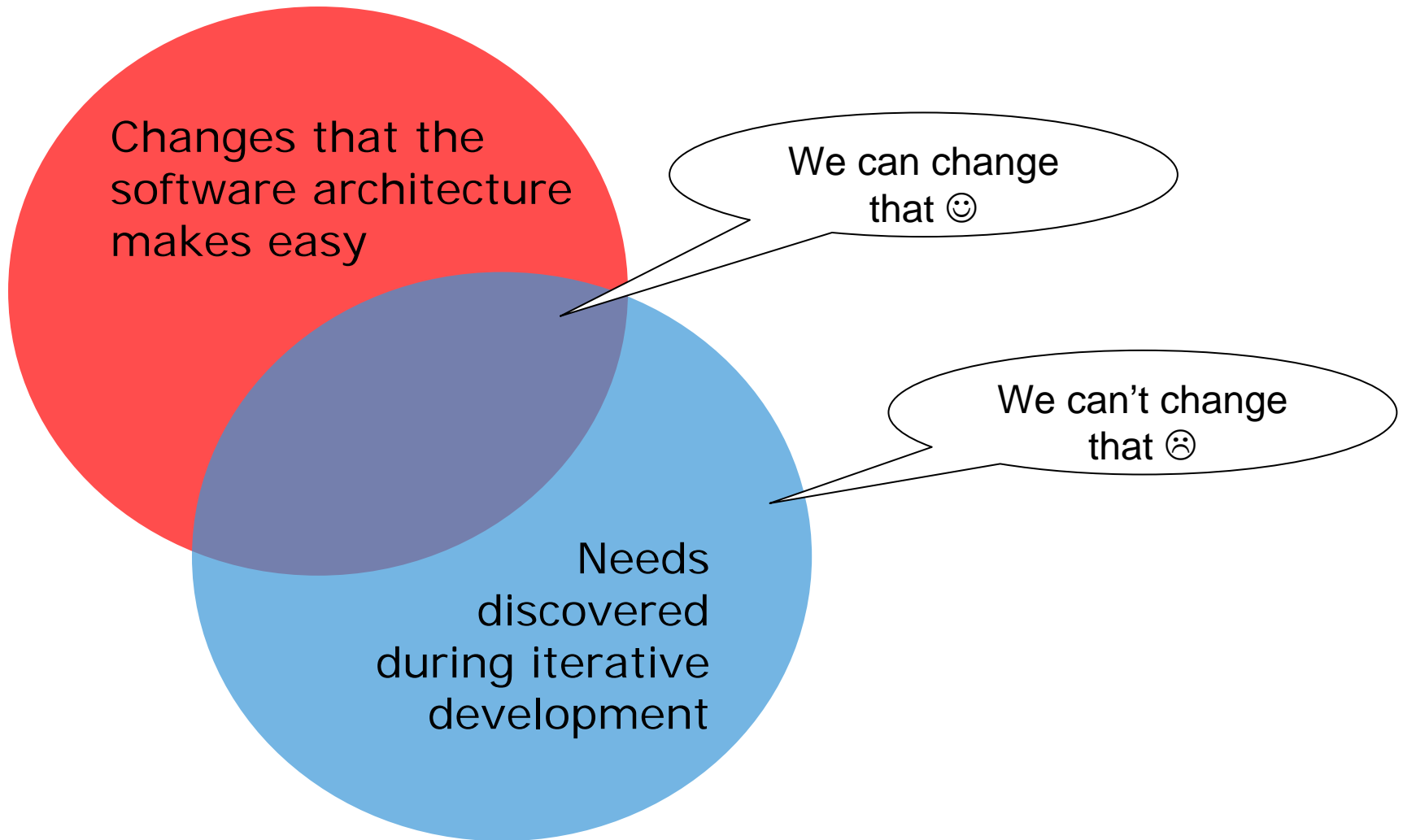
- PhD, Cognitive Psychology, Carnegie Mellon, 1988
- Faculty, Carnegie Mellon, 1988-present



- Elsa Golden

- Commercial software developer, 1991-2003
- PhD student, HCI and Interdisciplinary Education Research, Carnegie Mellon, 2003-present

# What leads to “We can’t change that!” reactions?



# Usability-Supporting Architecture Pattern

- A brief **usability scenario** that describes the situation that the USAP is intended to solve.
  - For example, “The user issues a command then changes his or her mind, wanting to stop the operation and return the software to its pre-operation state.”
- A description of the **conditions** under which the USAP is relevant.
  - For example, “A user is working in a system where the software has long-running commands, i.e., more than one second.”
- A characterization of the **user benefits** from implementing the USAP.
  - For example, “Cancel reduces the impact of routine user errors (slips) by allowing users to revoke accidental commands and return to their task faster than waiting for the erroneous command to complete.”
- A description of the **forces** that impact the solution.
  - For example, “No one can predict when the users will want to cancel commands”
- An implementation-independent description of the solution, i.e., **responsibilities of the software**.
  - For example, one implication of the force given above is the responsibility that “The software must always listen for the cancel command.”

# Pattern context

- It is useful to distinguish USAPs from other patterns for software design and implementation.
  - USAPs are not user interface patterns, that is, they do not represent an organization or look-and-feel of a user interface
  - Nor are USAPs structural software architecture patterns like Model-View-Controller (MVC)
  - USAPs are patterns of software responsibilities that can be applied to any structure. As such, they can be applied to any legacy architecture and can support the functionality called for in UI patterns

# 19 general usability scenarios

1. Progress feedback
2. **Warning/status/alert feedback\***
3. Undo
4. Canceling commands
5. **User profile\***
6. Help
7. Command aggregation
8. Action for multiple objects
9. Workflow model
10. Different view
11. Reuse of information
12. Maintaining compatibility with other systems
13. Navigating within a single view
14. Recovering from failure
15. Identity management
16. Comprehensive search
17. Supporting internationalization
18. Working at the user's pace
19. Shortcuts

\* Processed in CMU/ABB collaboration project

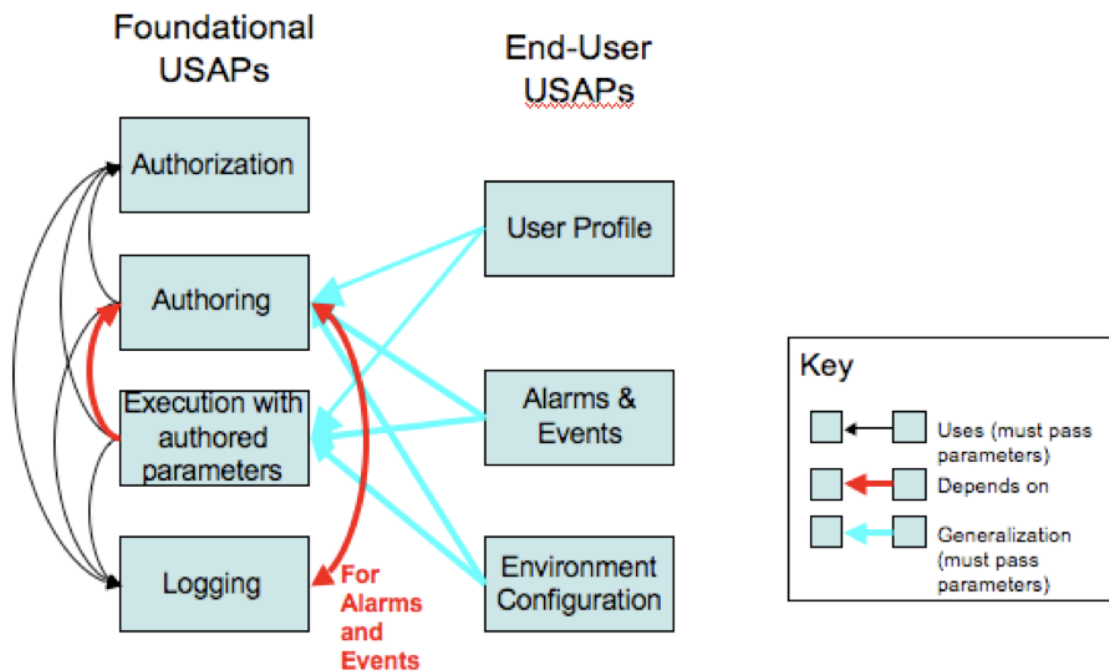
# A pattern language for USAPs

- ABB research team constructed “Alarm & Events” USAP
- CMU research team constructed “User Profile” and “Environment configuration” USAP
- Enormous amount of redundancy in the responsibilities necessary for a good solution
  - Negative reaction from industry software architects to the amount of information
- Both teams had grouped the responsibilities into similar categories
  - Led us to construct a pattern language that defines the relationships between USAPs

# CMU/ABB Research Breakthrough

- Two types of USAPs; Foundational and End-User USAPs
  - Foundational USAPs do not have the same six portions as the End-User USAPs
  - The responsibilities in the Foundational USAPs are parameterized where the parameters reflect those aspects of the End-User USAPs that differ

## Relationship between USAPs



# Prototype tool - responsibilities

- Implemented by ABB SECRC

The screenshot shows a Microsoft Internet Explorer browser window titled 'USAP Usability Test - Microsoft Internet Explorer'. The address bar shows the URL 'http://www.abb.com/...'. The page content is as follows:

**ABB Corporate Research**

Main Page

Authoring

Description

Responsibilities

- Create a specification
- Save a specification
- Modify a specification
- Delete a specification
- Exit the authoring system

Execution with authored parameters

Description

Responsibilities

- Access the appropriate specification
- Use specified parameters
- Additional responsibilities

Logging

Description

Responsibilities

- Specify the items to be logged

**USABILITY-SUPPORTING ARCHITECTURE PATTERN: AUTHORIZING**

**RESPONSIBILITIES**

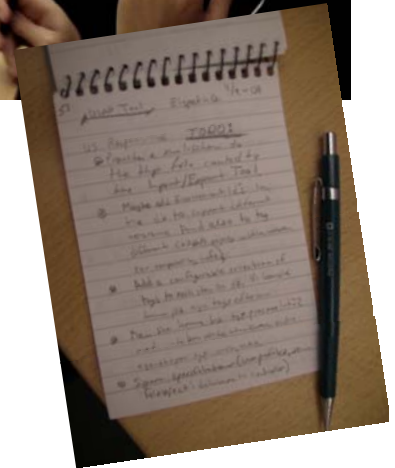
<input type="checkbox"/> <b>AUTHORIZING</b>		
<input type="checkbox"/> AU.1. Create a specification		
<input checked="" type="checkbox"/> AU.1.1 The system must provide a way for an authorized author to create a [User Profile, Configuration description, Conditions for Alarms, Events and Alerts].		
<a href="#">Show Rationale</a>		
<a href="#">Show Implementation Details</a>		
User Profile USAP: User Profile		<input type="radio"/> Not yet considered <input type="radio"/> Must modify architecture <input checked="" type="radio"/> Architecture addresses this <input type="radio"/> Not applicable <input checked="" type="checkbox"/> Discuss status of responsibility
Environment Configuration USAP: Configuration Description		<input type="radio"/> Not yet considered <input checked="" type="radio"/> Must modify architecture <input type="radio"/> Architecture addresses this <input type="radio"/> Not applicable <input type="checkbox"/> Discuss status of responsibility
Alarms, Events and Alerts USAP: Conditions for Alarms, Events and Alerts		<input type="radio"/> Not yet considered <input checked="" type="radio"/> Must modify architecture <input type="radio"/> Architecture addresses this <input type="radio"/> Not applicable <input type="checkbox"/> Discuss status of responsibility

# User tests

- Västerås, Sweden, 10-11 September 2008
  - The prototype tool was designed for the FM architecture
    - New architecture, very early stage of design
  - Single software architect from 800xA-system
    - The prototype tool was not designed for this architecture
    - Mature architecture, 10 years old

# User tests result

- FM architects
- ~20 min per responsibility, on average
- Generated 4 pages of ToDo items with the tool, 4 pages of hand-written notes detailing what to do
- Both FM and 800xA-system architects very positive
- FM architects want more NOW, as they are designing their new architecture
- U1: *“We have discussed lots of internal stuff in the system but this gave us some picture of what the user is going to see.”*
- U2: *“And that is things that we were not going to get that input, until very late in the design process, if we hadn’t used this tool now. So it was good to have these points of view come in this early.... I think we have identified at least a couple of new subsystems.”*
- U1: *“Yes. And some shortcomings of the previous design.”*
- U2: *“Yeah.”*



# Future work

Value to ABB

- Expand work to more BU's
  - Create more USAPs
  - Incorporate additional qualities
  - Deliver a web-based tool for global delivery

Researchers involved  
paper documents  
(CMU alone)

Architects, co-located  
Interactive prototype  
(ABB+CMU)

Globally-distributed  
teams create and use  
quality responsibilities in  
Web-based tool

**ABB**