How do you know?

Felix Bachmann
NW-MODE'09

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Software Engineering Institute

Federally funded Research and Development Center

Created in 1984

Under contract to Carnegie Mellon University

**Mission:** Improve the practice of software engineering
How do you know?

... you made the right choice
Agenda

To answer this question we need:

- An understanding of the architecture
- A measure
- The tool support

- How do you know? Bachmann, NW-MoDE 2009

© 2009 Carnegie Mellon University
Understanding the Architecture
The Dilemma of the Architect – Architecture Decision

Initial architecture may look like this

Architect decides to make the architecture better

Such as this one … or this one …

A view of possible architectures

Architecture

Decision
The Dilemma of the Architect – 2

And the process repeats …

Until (hopefully) a solution is found

A view of possible architectures
The Dilemma of the Architect – 3

How do you know? Bachmann, NW-MoDE 2009

© 2009 Carnegie Mellon University

... but there are many more architectures that have not been explored!

A view of possible architectures

... or the project runs out of time!

... and the perfect solution might be there explored!
The space of possible architectures for even a simple example is huge

It is like finding the needle in a haystack
… infinite possibilities
What do we, the software engineering community, do to help the architects? We create methods and tools to manage the complexity.

You, architect, you make the decisions and we help you to write it done

- Modeling tools that allow to create huge models
  - to model something that has a high probability to be wrong
- Refactoring tools that make it easy to fix mistakes
  - and therefore support making mistakes.

Where are the methods and tools to help the architects to make the right decisions; to effectively explore the design space?
Creating a design means breaking up the system into some number of pieces that are related to each other in some way, so that the pieces and their relations exhibit desired properties.

What is the criteria to break up an element into more pieces?

Why is A1 + A2 better than just A?
Separation of Concerns

Different functions usually have different quality attribute properties.
We separate functions with different properties to create structures that support those properties.

• e. g. If some functions need to be changed then they should be separated from those functions that do not change.

• If some functions always change together, then they should be located together

We are very familiar with this concept when it comes to make a design changeable.

The same concept applies to all other quality attributes too.

• e. g. Even if two functions are always changed together, but they have different security requirements, then they should be separated.
Separation of Concerns

The responsibilities F1 and F2 have different quality attribute properties (M6) than F3 and F4 (M1).

Separation of those functions enables a better design.
Challenge – Composition vs. Decomposition

Would it be possible to create a bottom-up approach to design?

Instead on decomposing a system into smaller pieces, can we do the following:

- Collecting all the functions of a system,
- Describing the dependencies between them,
- Assigning the quality attribute properties to them,
- Use the differences in the quality attribute properties to group the functions into bigger pieces?
The Architecture Design Process

An architecture design follows (should, really!) this process:

1. Create a measurable specification of quality attribute requirements that need to be supported by the architecture
2. Evaluate the quality attribute properties of the current architecture elements if they fulfill those requirements
3. If not, add structures to the architecture to improve and repeat step 2
4. If yes, Lucky you! You are done.

As simple as this may sound, it creates a huge problem …
Which is the Better Architecture?

When security is a concern

How about modifiability?

Architecture 1

Architecture 2
Is the Architecture Modifiable?

Change this!  Change this!

object Five Start

B

C

A

D

E

object Five Start - dependencies

B

C

A

D

E
Is the Architecture Modifiable?

Change this!

UML isch …
Obviously in the first diagram necessary information was missing. Step by step we added more information until we were able to come to a decision. The kind of information we have to add depends on the quality attribute we are focussing on.
Quality Attribute Information – 2

Every quality attribute has its own structures to influence the quality attribute properties of the architectural elements.

For example **security**:  
- The number of connections – the less the better (zero connections the best!)  
- The grouping of components into security areas

For example **modifiability**:  
- The localization of changes – the less places the better  
- The abstractions when accessing a component – the higher the abstraction of the interface the less likely changes will be visible outside the component.  
- Indirections – prevent changes from rippling through the architecture
Requirements

To design a system we need:

- The functional requirements
- The dependencies between them
- The quality attribute requirements

No requirement specification lacks the enumeration of functional requirements. But what about quality attribute requirements?

Let us talk a little bit more about them …
Quality Attribute Scenarios

Quality attributes have to be specified \textit{measurably} using quality attribute scenarios.

A fully specified quality attribute scenario consists of six parts:

- \textbf{Stimulus} – event that is effecting the system
- \textbf{Response} – activity as a result of the stimulus
- \textbf{Source of stimulus} – The entity that generated the stimulus
- \textbf{Environment} – the condition under which the stimulus occurred
- \textbf{Artifact stimulated} – the artifact that was stimulated
- \textbf{Response measure} – the measure by which the system’s response will be evaluated
Quality Attribute Scenarios

Example Availability Scenario:

An unanticipated external message is received by a process during normal operation. The process informs the operator of the receipt of the message and the system continues to operate with no down time.

For more information:
Software Architecture in Practice, 2nd Edition

How do you know? Bachmann, NW-MoDE 2009
© 2009 Carnegie Mellon University
Specify what functions are affected by what scenario.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Function</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Scenario 1</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Scenario 2</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Performance Scenario 3</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Scenario 4</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 1</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 2</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 4</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 5</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Quality Attribute Properties

What property has to be achieved by a function to support the quality attribute scenario?

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Function</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Scenario 1</td>
<td></td>
<td>Max(120ms)</td>
<td></td>
<td></td>
<td>Max(10ms)</td>
<td></td>
</tr>
<tr>
<td>Performance Scenario 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Jitter(&lt;1ms)</td>
</tr>
<tr>
<td>Performance Scenario 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Scenario 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Concurrency</td>
</tr>
<tr>
<td>Modifiability Scenario 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 2</td>
<td></td>
<td></td>
<td></td>
<td>Effort(5h)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Effort(1w)</td>
</tr>
<tr>
<td>Modifiability Scenario 4</td>
<td></td>
<td></td>
<td></td>
<td>Level 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifiability Scenario 5</td>
<td></td>
<td></td>
<td></td>
<td>location</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How do you know? Bachmann, NW-MoDE 2009

© 2009 Carnegie Mellon University
Quality Attribute Properties

After we know what the quality attributes of the functions need to be:

*How do you find out what the quality attribute properties of the current architecture are?*
The Measure

Quality Attribute Models
If this is all we have from an architecture, what can we say about its modifiability properties?

To reason about modifiability, we need some information about:

- How difficult is it to change an element?
- How likely it is that a change becomes visible to dependent elements?
From Architecture to Quality Attribute Model – 2

What we really need is something like this:

Architecture

Propagation of change

Interpretation

Modifiability Model

Interpretation Function

An interpretation function extracts the necessary information from a design and creates a quality attribute model from it.

Every quality attribute model requires its own interpretation function.

The design must include the necessary information otherwise the quality attribute model cannot be created.
With this simple model we can now estimate the average cost of certain changes:

Cost when changing A:
\[ A_c = 2.0 \]

Cost when changing B:
\[ B_c = 4.0 + 0.7 \times 2.0 = 5.4 \]

Cost when changing C:
\[ C_c = 5.0 + 4.0 \times 0.7 + 2.0 \times 0.7 \times 0.7 + 2.0 \times 0.2 = 9.18 \]
Are these numbers correct? Most likely not!

\[ A_c = 2.0 \]

\[ B_c = 4.0 + 0.7 \times 2.0 = 5.4 \]

\[ C_c = 5.0 + 0.7 \times 4.0 + 2.0 \times 0.7 \times 0.7 + 2.0 \times 0.2 = 9.18 \]

... but this model behaves like a typical design:

- Nothing depends on A, so changing A is not that expensive.
- The strong dependencies between CB and BA make changes in C ripple through the system (~50% chance for a change of A).
- Everything depends on C, so changing C is expensive.
Improvement Suggestions

The model also gives hints to want needs to be changed in the architecture to increase the modifiability.

To make the results better we can:

- Reduce the cost
- Reduce the probabilities
- Change the structure

... if C has to be modifiable

Can we decrease the cost of C?

- Maybe separating the functions of C into the part what really needs to be changed and the rest
Structural Change

\[ C_{1c} = 2.0 + 2.0 \times 0.2 = 2.4 \]

\[ C_{2c} = 3.0 + 4.0 \times 0.7 + 2.0 \times 0.7 \times 0.7 + 2.0 \times 0.7 + 2.0 \times 0.7 \times 0.2 = 8.46 \]
Preventing the Ripple Effect

… if C has to be modifiable

Can we decrease the probability of propagation from C to B?

• Maybe by increasing the abstraction level of the interface of C
Example: Modifiability Model

\[ C_c = 5.0 + 4.0 \times 0.2 + 2.0 \times 0.2 \times 0.7 + 2.0 \times 0.2 = 6.48 \]
Tactics

Every transformation in the quality attribute model has a corresponding transformation in the design model.

We call this transformation a tactic.

Tactics are small transformation in a design to promote a single quality attribute.
Tactic – Separate Functions

How do you know? Bachmann, NW-MoDE 2009
© 2009 Carnegie Mellon University
**Tactic – Increase Abstraction Level**

How do you know? Bachmann, NW-MoDE 2009

© 2009 Carnegie Mellon University
Design Process

Putting it all together

Architecture

QA Models
Real World

Yes, the real world is much more complicated:

- A normal system approximately requires 5-10 quality attribute models
- Although tactics are designed to improve a single quality attribute, they have side effects on other quality attributes
- Designing is still a struggle with many decisions without really knowing if the architect is on the right track

… but all those decisions can be more based on current knowledge and tools can help to revise earlier decisions.
The Tool Support
ArchE – Who is ArchE?

Your friendly guide helping you navigate through the space of architectures

Requirements in various forms

Available knowledge

Designer

ArchE

System

Architecture

How do you know? Bachmann, NW-MoDE 2009
© 2009 Carnegie Mellon University
What does ArchE do?

ArchE is a tool designed to provide useful information about a current architecture to the architect to find a good solution for a given problem. The tool understands:

- The design process
- Quality attributes and their connection to software architectures

The architect provides the domain knowledge.

Don’t be fooled by this nice presentation! ArchE is a prototype, sorry.
The Principles of Architecture Design

Functional Requirements

Assigned to

Architecture n

Repeat until satisfied

Architecture n + 1

Interpretation

Assigned to

Quality Attribute
Models, like Performance, Modifiability, etc.

Assigned to

Evaluation

Tactic # 7

Quality Attribute Requirements

Assigned to

Satisfied

How do you know? Bachmann, NW-MoDE 2009
© 2009 Carnegie Mellon University
Example – Responsibilities with Dependencies

Dependency Structure Matrix (DSM) showing the initial responsibilities with their dependencies

<table>
<thead>
<tr>
<th>Responsibility</th>
<th>Attach to Model</th>
<th>Create User Profile</th>
<th>Handle User Interactions</th>
<th>Locate Services</th>
<th>Manage External Devices</th>
<th>Manage Itineraries</th>
<th>Modify User Profile</th>
<th>Query Data</th>
<th>Register Views</th>
<th>Save Data</th>
<th>Show Itineraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach to Model</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Create User Profile</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Handle User Interactions</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Locate Services</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Manage External Devices</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Manage Itineraries</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Modify User Profile</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Query Data</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Register Views</td>
<td>9</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Save Data</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Show Itineraries</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Scenario Entry

A scenario is a quality attribute requirement of a system and is described in six parts.

**Scenario Text:**

Adding a new feature requires a change to the storage format. The implementation changes required for the new format are completed in 3.5 days

<table>
<thead>
<tr>
<th>Part</th>
<th>Text</th>
<th>Type</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stimulus</td>
<td>Add a new feature</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source of stimulus</td>
<td></td>
<td>End user</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artifact</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>The implementation is complete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response measure</td>
<td>within 3.5 days</td>
<td>Cost Constraint</td>
<td>Days</td>
<td>3.5</td>
</tr>
</tbody>
</table>
Scenario responsibility mapping

How do you know? Bachmann, NW-MoDE 2009
Example – Initial Design

UML Class Diagram when assigning each responsibility to its own module (class)

That is what ArchE does if designer did not specify anything different
Scenario Status

Models are created and evaluated as soon as the necessary information is available. Scenarios are marked as satisfied or not.
Evaluation of Tactics

ArchE evaluates suggested tactics and provides information about their influence on the architecture for the designer.

<table>
<thead>
<tr>
<th>Tactic</th>
<th>Improved</th>
<th>Neutral</th>
<th>Deteriorated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding a new feature requires to change the storage format. The imple…</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A new variable to the user profile has to be added within 5 days of effort</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The system has to manage the external devices under normal load and …</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A view wishes to attach to the model under normal conditions and do so …</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The driver for a new external device has to be added by a developer wi…</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Suggested Tactics – Shared Resources

Architect can either accept the suggestion and let ArchE do the transformation or changes the architecture manually.
Modifiability Impacted After Performance Tactic

First column depicts if scenario is satisfied or not

Second column depicts change after tactic was applied
Pointing in the Right Direction

More architecture alternatives can be explored if

- Evaluation of possible architectures is faster
- recognition and elimination of conflicts between quality attributes requirements are done as soon as they appear – to not end up in a dead end

Tool support can help here:

ArchE explored 7 (simple!) architectures in 2 sec
Before and After

Initial design

After a few iterations

It’s true, the second design is better!