Requirements-based Software System Adaptation in Practice: the Zanshin Framework

Vítor E. Silva Souza

Federal University of Espírito Santo
Vitória – ES, Brazil

vitorsouza@inf.ufes.br
http://www.inf.ufes.br/~vitorsouza

This material has been licensed under a Creative Commons Attribution 3.0 Unported license.
A bit of context

• Work done at the **Univ. of Trento**, Italy, during my PhD:
  – Advisor: **John Mylopoulos**;
  – Closest **collaborators**: Yiqiao Wang (Toronto), William N. Robinson (Georgia State), Alexei Lapouchnian (Toronto);

• Now currently a professor at the **Federal University of Espírito Santo** (Ufes) – Vitória, ES – Brazil.
Adaptive systems and feedback control

Users should always succeed in scheduling meetings

Meeting properly scheduled?

Important meeting?

Context information

Indicators of requirement convergence

System requirements

Adaptation framework

Adaptation actions

Target System

Meeting not properly scheduled

Reconfigure the system, try again, etc.
The Zanshin approach
The Zanshin framework
Current implementation: OSGi / Eclipse

Existing application

1: create and retrieve in-memory model

2: begin(), success(), fail(), ...

Repository Service

3: life-cycle methods called

4: AwReqs check

5: AwReq state change

6: retrieve strategies specified for failed AwReq

Controller Service

Remote two-way communication

7 (opt): use reconfiguration

Adaptation Service

8: send evolution instructions

Reconfiguration Service

Monitoring Service

Rule Engine

core
Experiments

- Didactic: the Meeting Scheduler [ER’11, CSRD’12, PhD];
- Proof-of-concept: an Adaptive CAD, based on the LAS-CAD [SEfSAS2, CoopIS’12, A-CAD Tech. Rep., PhD];
- An external application: Russel Bjork’s ATM simulation [SEAMS’13a];
- Comparison with an architecture-based approach (Rainbow): the ZNN.com slashdot effect [SEAMS’13b].

In the remainder of this talk, we will see how Zanshin works (an overview) using the A-CAD as example.
Baseline: goal model with variability

- Ambulances arrive in 8 min
- Dispatching occurs in 3 min
- Incidents resolved in 15 min
- Fast arrival
- Fast dispatching
- Fast assistance

**Call taking**
- Resource data is up-to-date
- Resource mobilization
- Resource identification
- Map retrieval
- Incident update
- Monitor resources

**Resource data is up-to-date**
- Specify configuration of ambulances
- Confirm incident

**Resource mobilization**
- Inform stations/ambulances
- Get good feedback
- Provide route assistance
- Staff member assists via radio
- A-CAD assists via navigator

**Resource identification**
- Confirm incident
- Inform stations/ambulances
- Get good feedback
- Provide route assistance
- Staff member assists via radio
- A-CAD assists via navigator

**Monitor status of ambulances**
- Display status of ambulances
- Display status of ambulances
- Display exception messages
- Display departure/arrival messages
- Display exception messages
- Add to message queue

**Incidents resolved in 15 min**
- Monitor status of ambulances
- Display status of ambulances
- Display exception messages
- Display departure/arrival messages
- Display exception messages
- Add to message queue
Step 1. Awareness Requirements

Ambulances arrive in 8 min
- Fast arrival
- SuccessRate(75%) (AR3)
- not TrendDecrease(30d, 2) (AR4)

Incidents resolved in 15 min
- Fast dispatching
- Fast assistance
- Dispatching occurs in 3 min
- NeverFail (AR11)

Call taking
- Resource data is up-to-date
- Resource identification

Resource mobilization
- SuccessRate(90%) (AR12)

Incident update
- Monitor resources

Map retrieval
- Obtain map info manually

Resource data is up-to-date
- Specify configuration of ambulances
- Determine best ambulances

Resource mobilization
- Get good feedback

Inform stations / ambulances
- Provide route assistance

Driver knows the way
- Staff member assists via radio
- A-CAD assists via navigator

Gazetteer working and up-to-date
- Check the gazetteer
- Check paper map

Map retrieval
- Obtain map info manually

Incident update
- Replace ambulance
- Close incident

Abnormal page detected
Some AwReqs of the A-CAD

Aggregate
AwReq AR1 should succeed 90% of the time, considering month periods

Aggregate
QC Ambulances arrive in 8 min should have 75% success rate;

Trend
The success rate of that same QC should not decrease 2 months in a row

Basic
QC Dispatching occurs in 3 min should never fail;

Aggregate
Task Get good feedback should succeed 90% of the time

Basic
Goal Register call should never fail;
Instance (for this dispatch)

“If the quality constraint that prescribes ambulances should arrive in 8 min fails, reconfigure”.

Class (from now on)

“If MDTs fail to communicate position every 13 seconds, relax it to every 20 seconds”.

Reconfiguration

Evolution
Parameters, indicators and relations

Basic

QC Dispatching occurs in 3 min should never fail;

AR 11

1. Consider AwReqs as indicators of requirements convergence;

- LoA: Level of Automation;
- VP1: goal *Provide route assistance*;
- VP2: goal Map retrieval;
- VP3: goal Obtain map info manually;
- VP4: goal Update position of engaged ambulances;
- MST: Minimum Search Time.

2. Elicit system parameters that can be tuned at runtime;
Parameters, indicators and relations

3. Elicit the (qualitative) effect that changes in parameters have in indicators;

- $\Delta(\text{AR11}/\text{LoA}) > 0$
- $\Delta(\text{AR11}/\text{VP1}) < 0$
- $\Delta(\text{AR11}/\text{VP2}) < 0$
- $\Delta(\text{AR11}/\text{VP3}) < 0$
- $\Delta(\text{AR11}/\text{VP4}) < 0$
- $\Delta(\text{AR11}/\text{MST})[0, 180] > 0$

4. Refine these relations, e.g., establishing an order of effect, if possible;

- $|\Delta(\text{AR11}/\text{VP2})| >$
- $|\Delta(\text{AR11}/\text{LoA})| >$
- $|\Delta(\text{AR11}/\text{VP3})| >$
- $|\Delta(\text{AR11}/\text{VP1})| >$
- $|\Delta(\text{AR11}/\text{VP4})| >$
- $|\Delta(\text{AR11}/\text{MST})|$

5. Choose a reconfiguration algorithm...
Qualia: customizable/extensible algorithm

1. Choose param(s)
2. Decide inc /dec value
3. Increment / decrement
4. Wait
5. Evaluate
6. Learn
7. Are we done?
8. Reassess

Yes: Next
No: Choose param(s)
Step 3. Evolution Requirements

- Reconfiguration – solution space:
  - “If dispatching doesn't occur in 3 minutes, reduce the time spent searching for duplicates”;
- Evolution – problem space:
  - “If the goal of registering a call is not achieved successfully, retry in 5 seconds or disable caller location detection”;
  - “If the assumption that resource data is up-to-date is false, delegate the solution to a staff member”;
  - “If MDTs fail to communicate position every 13 seconds, relax it to every 20 seconds”.

September 2013
EASSy 2013 @ Shonan Village Center, Kanagawa, Japan
## Adaptation specification

<table>
<thead>
<tr>
<th>AR</th>
<th>Adaptation strategies (EvoReqs)</th>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR5</td>
<td>1. Delegate(“Staff Member”).</td>
<td>1. Always</td>
</tr>
<tr>
<td>AR8</td>
<td>1. RelaxReplace(D_MDPs_20s); 2. RelaxReplace(AR8, AR8_45s); 3. RelaxReplace(AR8_45s, AR8_30s); 4. Retry(6000); 5. Reconfigure(∅ [Immediate resol.]).</td>
<td>1. Once per session; 2. Once per session; 3. Once per session; 4. 3x per session; 5. Always.</td>
</tr>
</tbody>
</table>

Operationalized by an algorithm.
Try it yourself!

https://github.com/sefms-disi-unitn/Zanshin

Source code for the Zanshin framework, simulations console and the “Zanshin-managed” ATM software.

https://github.com/sefms-disi-unitn/Zanshin/wiki

In the wiki, instructions on how to download Zanshin, how to run its simulations, how to create your own models and simulations and how to run the ATM experiment.
Current / future directions

- Further development of the prototype framework;
- Revision of its base model (ontology of requirements);
- LawReqs: studying the relation between regulatory compliance and adaptive systems [SEAMS’13c];
- Move towards architecture: model-driven approach;
- Handling the (many) limitations of Zanshin:
  - Too much responsibility on the system developer (integration) and analysts (consistency, correctness);
  - No support for legacy / 3rd party systems;
  - No integration with domain-specific models;
  - Qualia: empty procedures.
References


Nemo is supported by Brazilian agency FAPES (www.fapes.es.gov.br) through the PRONEX grant #52272362 and others.

This work has been supported by the ERC advanced grant 267856 “Lucretius: Foundations for Software Evolution” (April 2011 – March 2016) – http://www.lucretius.eu.
**“Vanilla” requirements**

<table>
<thead>
<tr>
<th>#</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ-1</td>
<td>The system shall allow staff to register calls they receive from citizens;</td>
</tr>
<tr>
<td>REQ-2</td>
<td>The system shall, whenever possible, detect the location of the caller and associate it with the call registry (public phones have associated locations, cell phones might be triangulated, etc.);</td>
</tr>
<tr>
<td>REQ-3</td>
<td>The system shall allow staff to dismiss calls as non-emergencies;</td>
</tr>
<tr>
<td>REQ-4</td>
<td>The system shall assist staff in identifying, through the information from the call, if it refers to an open incident in the system;</td>
</tr>
<tr>
<td>REQ-5</td>
<td>The system shall allow staff to assign calls to open incidents as duplicates or create new incidents for calls;</td>
</tr>
</tbody>
</table>
Initial goal model

- **Ambulances arrive in 8 min**
  - Call taking
    - Resource data is up-to-date
    - Resource identification
  - Generate optimized dispatching instructions
  - Incidents resolved in 15 min
    - Fast arrival
    - Incident update

- **Fast arrival**
  - Call taking
    - Resource data is up-to-date
    - Resource identification
  - Generate optimized dispatching instructions
  - Incidents resolved in 15 min
    - Fast arrival
    - Incident update

- **Up to 1500 calls per day**
  - Register call
    - Search for duplicates
    - Detect caller location
  - Input emergency information
  - Confirm call
    - Assign to incident
    - Specify configuration of ambulances
  - Confirm incident
    - Confirm ambulances
    - Gazetteer working and up-to-date
  - Determine best ambulances
  - Inform stations / ambulances
    - Monitor resources
      - MDTs communicate position
        - Crew members use MDTs properly
          - Monitor status of ambulances
            - Display status of ambulances
            - Display exception messages
            - Display departure/arrival messages
  - Replace ambulance
  - Close incident
    - Fast dispatching

**Key:**
- **Goal**
- **Task**
- **Domain assumption**
- **Soft-goal**
- **Refinement**
- **Quality constraint**

**September 2013**

EASSy 2013 @ Shonan Village Center, Kanagawa, Japan

23