On the Composition and Reuse of Viewpoints across Architecture Frameworks

Rich Hilliard
Freelance software systems architect

Ivano Malavolta, Henry Muccini, Patrizio Pelliccione
Department of Engineering, Computer Science and Mathematics, University of L’Aquila, Italy

www.henrymuccini.com, henry.muccini@di.univaq.it, @muccinihenry

@WICSA/ECSA 2012, Helsinki, August 2012
Stakeholders' concerns can vary tremendously (and change over time), depending on:

- the nature of the system
- project-specific constraints
- organizational constraints
- the application domain
- …
Stakeholders' concerns can vary tremendously (and change over time).

We require a way to capture stakeholder's specific architectural concerns.

It is common practice to use multiple views and viewpoints to deal with different concerns.
Using multiple views has become standard practice in industry

- Based on a survey we conducted with 48 practitioners [Survey2012], and about the usage of ALs in industry
  - 85% uses multiple views

<table>
<thead>
<tr>
<th>Support for multiple architectural views</th>
<th>Useful in past projects</th>
<th>Blank</th>
<th>Useful for future projects</th>
<th>Don't know</th>
<th>Blank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-2 -1 0 +1 +2 No exp.</td>
<td></td>
<td>-2 -1 0 +1 +2 Don't know</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 2 4 11 18 2</td>
<td></td>
<td>9 1 0 4 5 27 10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Current issues

... but current AFs and ADLs are for the most part closed

I1  • difficult to reuse viewpoints when defining new frameworks across organizations and domains

I2  • The addition of new views (e.g., to frame a particular concern) is far from being systematic *

I3  • views consistency is still one of the harder problems in software architecture**

* 43% had to extend their ALs to add new views [Survey2012]
** Also remarked in “Just Enough Software Architecture”, George Fairbanks, 2010
ADLs and AFs

1) **Architecture Viewpoints:** define the contents of each architecture view;

2) **Architecture Frameworks (AFs):** coordinated set of viewpoints for use within a particular stakeholder community or domain of application (e.g., GERAM, TOGAF, MODAF);

3) **Architecture Description Languages (ADLs):** any mode of expression used in an architecture description. ADL provides model kinds selected to frame one or more concerns.
Goal of this work

To provide an **infrastructure** that enables to build **reusable** architecture frameworks by treating views, viewpoints, concerns as first-class entities.

MEGAF is an MDE approach to create new architecture frameworks by means of mechanisms:

i. to **store**, **retrieve**, and **combine** existing viewpoints, by properly selecting and reusing models previously defined and resident in MEGAF;

ii. to **define correspondences** among views, viewpoints, stakeholders, system concerns and their elements;

iii. to **enforce consistency** and **completeness checks** based on defined architectural relationships and rules among elements.
How to manage models that contains classes and other models?
A megamodel is a kind of model in which elements could represent and/or refer to models or metamodels [Bézivin et al., OOPSLA/GPCE 2004]

A megamodel specifies properties and rules governing model construction, including multiple models and metamodels

- Models and metamodels are first-class entities
- It offers also the possibility to specify relationships between them and to navigate them.
In MEGAF, a **megamodel is a repository of AD elements**

- Megamodels in combination with **weaving models** for coordinating sets of models;
- **The navigability and traceability extension.**

[Jouault et. al, ACM SAC 2010]
Solution within the MDE

- Stakeholders
  - stk

- Viewpoints
  - vp

- System Concerns
  - sc

- Model Kinds
  - mk

- Architectural Languages
  - c1

- Correspondences
Information systems for public transportation (BOA)

**Viewpoints**: structural, behaviour, web services

**ADLs**: Diasuite, UML

**Extensions**: REST services metamodel, LTS

**Stakeholders**: sw architect, end-user, developer, sys eng.
42010 AF definition
Work Done...

Definition of the GMM4SA metamegamodel, fully compliant to the ISO/IEC/IEEE 42010

- Each megamodel conforming to it must satisfy those relationships in order to be valid:
  - definition of conformance of an AF to the 42010
  - definition of conformance of an AD to an AF
  - definition AF correspondence rules

Specification of the model weaving and composition mechanisms

Use of the AM3 megamodel management component (in the AMMA platform) to record all available resources, acting as an MDE repository.
MEGAF classes of users

MD engineers
- define the domain megamodel (conforming to the GMM4SA) and creates viewpoints
- define correspondence rule in OCL

Software architects
- create AFs by easily using elements in the megamodel
- create ADs conforming to the AFs
  - create new view and import its model (done with existing ADLs) into MEGAF
Future Work

- Advanced searches
- Overlapping viewpoints management
- Usability and GUI
- Extension and customization of repository elements
- AF extensions can create problems to the corresponding AD

- Application to industrial projects
Reuse, compose, and extend ADLs and AFs instead of creating new ones
Our solution

MEGAF: a model-driven infrastructure for building reusable and extensible architecture frameworks

DuALLy: an automated approach for ADLs interoperability

byADL: an approach to adapt and customize existing ADLs

Composed AF generated in MEGAF
Extended/customized ADL generated in byADL
BPMN
Darwin/FSP

SA UML profiles
pivot metamodel \((A_0)\)
other ADLs

ACME
AADL
xADL

VTx
VP2
MK2
VT1
St1

Generated in MEGAF
Tool Support

- DUALLy
- byADL
- other engines
- MEGAF

- AMMA
  - AM3
  - AMW
  - ATL

- EMF
Automation

megaf.di.univaq.it
- Preliminary prototype in Eclipse, using megamodeling techniques

dually.di.univaq.it
- Prototype in Eclipse, using model-driven engineering techniques

byadl.di.univaq.it
- Prototype in Eclipse, using model-driven engineering techniques
On the Composition and Reuse of Viewpoints across Architecture Frameworks

Rich Hilliard
Freelance software systems architect

Ivano Malavolta, Henry Muccini, Patrizio Pelliccione
Department of Engineering, Computer Science and Mathematics, University of L’Aquila, Italy

www.henrymuccini.com, henry.muccini@di.univaq.it, @muccinihenry