Collaborative Construction of Telecommunications Services. An Enterprise Architecture and Model Driven Engineering Method

Vanea Chiprianov

Supervisors: Yvon Kermarrec
Siegfried Rouvrais

Ph.D. defense 16 January 2012
Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- State of the art: Advantages and limitations
- Solution/Contribution:
  1. Telecom service construction process
  2. Tool building process
  3. Software tools for telecom service construction
- Complete case study on a multimedia conferencing service
- Conclusion and perspectives
"The offering of telecommunications for a fee directly to the public, or to such classes of users as to be effectively available directly to the public, regardless of the facilities used."

[Federal Communications Commission, 1996]

**Broad definition:**
- Any content of telecommunications
  - e.g. voice, video
- Several types of facilities
  - e.g. circuit or packet switched network

**Examples:**
- Multimedia conferencing service, phone call, et al.
Context: Telecom service stakeholders
Context: Telecom service stakeholders

Stakeholders cf. [Haalstrand, 1994]
Context: Telecom service stakeholders

Stakeholders cf. [Haalstrand, 1994]
Context: Telecom service stakeholders

Stakeholders cf. [Haalstrand, 1994]
**Context: Telecom service life-cycle**

[Berndt, 1994]

<table>
<thead>
<tr>
<th>Need Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
</tr>
<tr>
<td>Definition</td>
</tr>
<tr>
<td>Specification</td>
</tr>
<tr>
<td>Verification</td>
</tr>
<tr>
<td>Development</td>
</tr>
<tr>
<td>Validation</td>
</tr>
<tr>
<td>Conformance Testing</td>
</tr>
<tr>
<td>System Testing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
</tr>
<tr>
<td>Activation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deployment</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Withdrawal</th>
</tr>
</thead>
</table>

- Provider Control and Operation
- Customer Control
- Subscription
  - Authorization
    - Service
    - Instance
    - End-User
    - Usage
    - Access
    - Interact
    - Exit
- Bar
- Cancellation
- Deactivation
- Removal

**Collaborative construction of telecommunications services**
Context: Our focus on telecom service life-cycle

<table>
<thead>
<tr>
<th>Need Capture</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Definition</td>
</tr>
<tr>
<td></td>
<td>Specification</td>
</tr>
<tr>
<td></td>
<td>Verification</td>
</tr>
<tr>
<td></td>
<td>Development</td>
</tr>
<tr>
<td></td>
<td>Validation</td>
</tr>
<tr>
<td></td>
<td>Conformance Testing</td>
</tr>
<tr>
<td></td>
<td>System Testing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Construction</th>
<th>Service Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Service Developer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Installation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
<th>Subscription</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider Control and Operation</td>
<td>Authorization</td>
</tr>
<tr>
<td></td>
<td>Service Instance</td>
</tr>
<tr>
<td></td>
<td>End - User Usage</td>
</tr>
<tr>
<td></td>
<td>Access</td>
</tr>
<tr>
<td></td>
<td>Interact</td>
</tr>
<tr>
<td></td>
<td>Exit</td>
</tr>
<tr>
<td>Customer Control</td>
<td>Bar</td>
</tr>
<tr>
<td></td>
<td>Cancellation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Withdrawal</th>
<th>Deactivation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Removal</td>
</tr>
</tbody>
</table>
Context: Main challenges in telecom service life-cycle

- Convergence of the traditional circuit-switched networks with the packet-switched ones
  - Services as software
- Increased involvement of the End User in the telecom service life-cycle
- Market deregulation
  - Increased competition
- Concept-to-market time, quality of service, cost, et al.
Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- State of the art: Advantages and limitations
- Solution/Contribution:
  1. Telecom service construction process
  2. Tool building process
  3. Software tools for telecom service construction
- Complete case study on a multimedia conferencing service
- Conclusion and perspectives
Research question

- How to reduce telecommunications service construction time while affecting non-negatively other parameters (e.g. Cost, QoS, QoE)?

**RQ 1 Construction process**

**RQ 2 Software tools**

**RQ 3 Tool building process**
Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- **State of the art: Advantages and limitations**
- Solution/Contribution:
  1. Telecom service construction process
  2. Tool building process
  3. Software tools for telecom service construction
- Complete case study on a multimedia conferencing service
- Conclusion and perspectives
State of the art

Obs.: existing literature focuses on software tools.

Requirements of Service Providers and Developers for Service Creation Environment (SCE):

- Req 1: An overall model
- Req 2: Domain specificity
- Req 3: Rapid prototyping
- Req 4: Collaborative support
- Req 5: Early verification/simulation
- Req 6: Integration
- Req 7: Reuse
- Req 8: Wide range of services
- Req 9: Easy evolution of services

[Blum, 2009], [Haalstrand, 1994], [Khlifi, 2008], [Kosmas, 1997], [Yelmo, 2008]
State of the art: Service Creation Environments categories

- Cat 1: SIP-dependent
- Cat 2: SIP-independent
- Cat 3: NGN composition
- Cat 4: Parlay X
- Cat 5: Web mash-up
- Cat 6: Hybrid

For Next Generation Network (NGN) (circuit-switched networks):
- Session Initiation Protocol (SIP) [Rosenberg, 2002]

For Web (packet-switched networks):
- Parlay X - [ETSI, 2006]
### State of the art: Comparison of SCEs

<table>
<thead>
<tr>
<th>Req</th>
<th>Feature</th>
<th>NGN (SGP)</th>
<th>Web</th>
<th>Cat 6 Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An overall model</td>
<td>Cat 1</td>
<td>Cat 4</td>
<td>Cat 6 Hybrid</td>
</tr>
<tr>
<td>2</td>
<td>Domain specificity</td>
<td>Cat 2</td>
<td>Cat 5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Rapid prototyping</td>
<td>Cat 3</td>
<td>Cat 6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Collaborative support, Req</td>
<td></td>
<td>Cat 6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Early verification/simulation, Req</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Integration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Reuse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Wide range of services</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Easy evolution of services</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### State of the art: Comparison of SCEs

<table>
<thead>
<tr>
<th>Req</th>
<th>Requirement</th>
<th>NGN</th>
<th>Web</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An overall model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Domain specificity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Rapid prototyping</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Collaborative support, Req 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Early verification/simulation, Req 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Reuse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Wide range of services</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Easy evolution of services</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NGN:
- **Cat 1**: SIP - independent
- **Cat 2**: SIP - independent
- **Cat 3**: NGN composition
- **Cat 4**: Parlay X
- **Cat 5**: Web mash-up
- **Cat 6**: Hybrid

Note: not addressed by any of the categories.
### State of the art: Comparison of SCEs

<table>
<thead>
<tr>
<th>Req</th>
<th>Requirement</th>
<th>NGN</th>
<th>Web</th>
<th>Cat 6 Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>CoAdR model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>An overall model</td>
<td>Cat 1 SIP-independent</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain specificity</td>
<td>Cat 2 SIP-independent</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rapid prototyping</td>
<td></td>
<td>Cat 3 NGN composition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Collaborative support, Req 5 Early verification/simulation, Req 6 Integration</td>
<td></td>
<td>Cat 4 Parlay X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reuse</td>
<td></td>
<td>Cat 5 Web mash-up</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Req 9 Easy evolution of services</td>
<td>Cat 6 Hybrid</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Req 1**: An overall model
- **Req 2**: Domain specificity
- **Req 3**: Rapid prototyping
- **Req 4**: Collaborative support, Req 5 Early verification/simulation, Req 6 Integration
- **Req 7**: Reuse
- **Req 8**: Wide range of services
- **Req 9**: Easy evolution of services

Related to the **individual** architecture of SCEs.
State of the art: Comparison of SCEs

<table>
<thead>
<tr>
<th>Requirement</th>
<th>NGN</th>
<th>Web</th>
<th>Cat 6 Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req 1 An overall model</td>
<td>not addressed by any of the categories</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Req 2 Domain specificity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Req 3 Rapid prototyping</td>
<td>- -</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Req 4 Collaborative support, Req 5 Early verification/simulation, Req 6 Integration</td>
<td>related to the individual architecture of SCEs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Req 7 Reuse</td>
<td>- -</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Req 8 Wide range of services</td>
<td>- -</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Req 9 Easy evolution of services</td>
<td>- -</td>
<td>+</td>
<td>++</td>
</tr>
</tbody>
</table>

bad -- 0 + ++ good
### State of the art: Comparison of SCEs

<table>
<thead>
<tr>
<th>Req 1</th>
<th>Req 2</th>
<th>Req 3</th>
<th>Req 4</th>
<th>Req 5</th>
<th>Req 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req 1</td>
<td>An overall model</td>
<td>not addressed by any of the categories</td>
<td>related to the <strong>individual</strong> architecture of SCEs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Req 2</td>
<td>Domain specificity</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>0</td>
</tr>
<tr>
<td>Req 3</td>
<td>Rapid prototyping</td>
<td>- -</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Req 4</td>
<td>Collaborative support, Req 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Req 5</td>
<td>Early verification/simulation, Req 6 Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Req 6</td>
<td>Reuse</td>
<td>- -</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Req 7</td>
<td>Wide range of services</td>
<td>- -</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Req 8</td>
<td>Easy evolution of services</td>
<td>- -</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Cat 1** SIP - independent
**Cat 2** SIP - independent
**Cat 3** NGN composition
**Cat 4** Parlay X
**Cat 5** Web mash-up
**Cat 6** Hybrid

**Page 10/55**

**Chiparianov Ph.D. defense**

**Collaborative construction of telecommunications services**
State of the art: Major limitations

- No category of SCE addresses *Req 1: An overall model*

- Choice between either:
  - *Req 2: Domain specificity* XOR
  - *Req 3: Rapid prototyping, Req 7: Reuse, Req 8: Wide range and Req 9: Easy evolution*
State of the art: Major limitations

- No category of SCE addresses **Req 1: An overall model**

  - Enterprise Architecture (EA)

- Choice between either:
  - **Req 2: Domain specificity XOR**
  - **Req 3: Rapid prototyping, Req 7: Reuse, Req 8: Wide range and Req 9: Easy evolution**
State of the art: Major limitations

- No category of SCE addresses *Req 1: An overall model*
  - Enterprise Architecture (EA)

- Choice between either:
  - *Req 2: Domain specificity* OR *AND*
  - *Req 3: Rapid prototyping, Req 7: Reuse, Req 8: Wide range and Req 9: Easy evolution*
  - Model Driven Engineering (MDE)
Enterprise Architecture

"A coherent whole of principles, methods and models that are used in the design and realization of the enterprise's organizational structure, business processes, information systems and infrastructure."

[Jonkers, 2006]

• Enterprise Architecture frameworks (e.g. TOGAF)
• Enterprise Architecture Modeling Languages (EAMLS) (e.g. ArchiMate)
Contributions of EA towards fulfilling SCE requirements

- Req 1: Overall model
- Req 6: Integration
- Req 7: Reuse
- Req 8: Wide range
- Req 9: Easy evolution

- Unified View
- ArchiMate inter-layer relations
- Layered architecture
- Separation from the platform
- TOGAF addresses the evolution of EAs
Model Driven Engineering (MDE)

A software development method which focuses on creating and exploiting domain models to simulate, estimate, understand, communicate and produce code.

[Gherbi, 2009]

- The Meta-modeling approach for language definition
- Model Transformations (model to model, code generation)
- Meta-tools
Contributions of MDE towards fulfilling SCE requirements

- Req 2: Domain specificity
- Req 3: Rapid prototyping
- Req 5: Early verification
- Req 6: Integration
- Req 7: Reuse
- Req 8: Wide range
- Req 9: Easy evolution

- Meta-models
- High automation
- Leveraging domain-specific tools
- Model Transformations
- Capture by meta-models
- Platform independence
- High automation
Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- State of the art: Advantages and limitations

Solution/Contribution:
1. Telecom service construction process
2. Tool building process
3. Software tools for telecom service construction

- Complete case study on a multimedia conferencing service
- Conclusion and perspectives
Solution: Telecom service construction process

Responsibilities of a modeler

1. Model
2. Test
3. Collaborate
4. Interoperate

[Kruchten, 2008]

Modeler = system/software architect/designer
Solution: Telecom service construction process

1. Model
2. Test
3. Collaborate
4. Interoperate
Solution: Telecom service construction process
Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- State of the art: Advantages and limitations

**Solution/Contribution:**

1. Telecom service construction process
2. Tool building process
3. Software tools for telecom service construction

- Complete case study on a multimedia conferencing service
- Conclusion and perspectives
Solution: Tool building process

1. Model
2.
3.
4.
Solution: Tool building process

1. Define MMcollab
2. Define MMi
3. Combine MMcollab with MMi
4. Define "display-surface" MM
5. Define semantics
6. Generate graphical editor
7. Generate code template-based
8. Reason on DR
9. Yes: Generate MTi,i+1
10. No: Lift to ONTOi
11. Enrich ONTOi
12. Enrich ONTOi+1
13. Transform to MM testing
14. Integrate results into MM
15. Validate DSML family
Solution: Tool building process

1. Define MMcollab
2. Define MMI
3. Transform to MM testing
4. Integrate results into MMI

Combine MMcollab with MMI

Define “display surface” MMI

Generate graphical editor
Generate code (template-based)

Define semantics

Reason on DR

Align ontologies

Generate MTU + 1

Validate DSML family

Describes the model from her viewpoint using her DSML

Interacts with the other Modelers

Tests the model

No
Test satisfactory?
Yes

Integrate constraints from other views

Imports the model into her DSML

Modeler_i

Constraints_{i+1}
Solution: Tool building process
Solution: Tool building process
Solution:
Tool building process

1. Define MMcollab
2. Define MMI
3. Combine MMcollab with MMI
   - Define "display-surface" MM
   - Generate graphical editor
4. Transform to MM testing
5. Integrate results into MM
6. Lift to ONTOi
7. Enrich ONTOi
8. Enrich ONTOi+1
9. Align ontologies
10. Generate MTi,i+1
11. Validate DSML family
    - Reason on DR
    - Generate code template-based
    - DR?
      - No
      - Yes
12. Align ontologies
13. Generate MTi,i+1
14. Validate DSML family
    - Ok?
      - No
      - Yes
Solution: Tool building process

4. Interoperate

1. Define MMcollab
2. Define MMI
3. Transform to MMtesting
4. Integrate results into MM
5. Combine MMcollab with MMI
6. Define "display-surface" MM
7. Define semantics
8. Generate graphical editor
9. Generate code template-based
10. Reason on DR
11. Align ontologies
12. Generate MTU\(i+1\)
13. Validate DSML family
14. No
15. Yes
16. Lift to ONTOI
17. Enrich ONTOI
18. Enrich ONTOI\(+1\)

Model\(i\)

Model\(i-1\)

Model\(i\):

1. Imports the model into her DSML
2. Describes the model from her viewpoint using her DSML
3. Interacts with the other Modelers
4. Tests the model
5. No
6. Yes
7. Test satisfactory?
8. Yes
9. No
10. Integrate constraints from other views
11. Constraints\(i+1\)

Model\(i\):
Solution: Tool building process

- Define MMcollab
- Define MMi
- Define "display-surface" MM
- Generate graphical editor
- Define semantics
- Generate code template-based
- Reason on DR
- Align ontologies
- Generate MTi,i+1
- Integrate results into MMi
- Lift to ONTOi
- Enrich ONTOi
- Enrich ONTOi+1
- Transform to MM testing
- Combine MMcollab with MMI
- Validate DSML family

DR? No Yes

Ok? Yes No
Collaborative construction of telecommunications services

Chiprianov Ph.D. defense

Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- State of the art: Advantages and limitations

**Solution/Contribution:**
1. Telecom service construction process
2. Tool building process
3. Software tools for telecom service construction

- Complete case study on a multimedia conferencing service
- Conclusion and perspectives
Solution: Software tools
Solution: Modeling with DSMLs

Domain Specific Modeling Language (DSML) = graphical language that offers

- expressive power focused on a particular domain,
- to visualize, specify, construct and document the artifacts of a software-intensive system.

[Booch, 2005], [Deursen, 2000]
Solution: Modeling with DSMLs

Telecom Archi (GPL Type licence) extension [http://archi.cetis.ac.uk/]
Solution: Modeling with DSMLs

- Meta-models: 46 concepts
- Code generation semantics: lyassAlloush, masters internship, 6months, 39.8 KB
- Graphical editor: 395.8 KB
Solution: Software tools

1. Define MMcollab
2. Define MMI
3. Transform to MM testing
4. Integrate results into MM
5. Combine MMcollab with MMI
6. Define "display surface" MM
7. Generate code (template-based)
8. Validate DSM family
9. No
10. Yes

1. Lift to ONTO
2. Enrich ONTO
3. Align ontologies
4. Reason on DR
5. Generate MTU + 1

1. Describes the model from her viewpoint using her DSML
2. Interacts with the other Modelers
3. Tests the model
4. Integrate constraints from other views
5. Yes
6. No
7. Test satisfactory?
8. Yes
9. No
10. Model

Collaborative construction of telecommunications services
Collaborative construction of telecommunications services

Solution: Testing through leverage of COTS

Components Off The Shelf (COTS) = "a commercially available or open source piece of software that other software projects can reuse and integrate into their own products"

[Torchiano, 2004]
Solution: Leveraging the COTS – Transform to MM testing

Excerpt of Xpand template for OPNET input model generation

Excerpt from the model of a conferencing service at the Telecom ArchiMate Technology layer

Xpand (OpenArchitectureWare) [Efftinge, 2006]

The static configuration of the conferencing service excerpt model for OPNET
Solution: Leveraging the COTS – Transform to MM testing

Excerpt of Xpand template for OPNET input model generation

Excerpt from the model of a conferencing service at the Telecom ArchiMate Technology layer [Efftinge, 2006]

The static configuration of the conferencing service excerpt model for OPNET

Code generation Model Transformation: Iyass Alloush, masters internship, 6 months, 122.4 KB
Solution: Software tools

1. Define MMcollab
2. Define MM
3. Collaborate
4. Transform to MM testing
5. Integrate results into MM

- Combine MMcollab with MM
- Define "display-surface" MM
- Define semantics
- Generate code template-based
- Generate code
- Reason on DR
- Align ontologies
- Generate MTL(i+1)
- Validate DSML family

- No
- Yes

- No
- Yes

- No
- Yes

- No
- Yes

- No
- Yes

Modeler_i

Interacts with the other Modelers

Describes the model from her viewpoint using her DSML

Tests the model

Integrate constraints from other views

Imports the model into her DSML

Model_i

Constraints_{i+1}
Solution: Collaborating by capturing and retrieving Decision Rationale

Decision Rationale DSML

Decision Rationale = the justification behind decisions, the reasoning that goes into determining the design of the artifact.

[Dutoit, 2006]
Solution: Defining a Decision Rationale

DSML

- Meta-model: 6 concepts
- Graphical editor: Adil Meribaa and Mosbah Lassoued, masters internships, 2 weeks, 108.8 KB
Solution: Software tools

1. Define MMcollab
2. Define MM
3. Transform to MM testing
4. Integrate results into MM

Combine MMcollab with MM

Define "display-surface" MM

Define semantics

Generate graphical editor

Generate code template-based

Reason on DR

Generate MTi+1

Validate DSML family

Lift to ONTDI

Enrich ONTDI

Enrich ONTDI+1

4. Interoperate

Imports the model into her DSML

Describes the model from her viewpoint using her DSML

Interacts with the other Modelers

Tests the model

Integrate constraints from other views

No

Test satisfactory?

Yes

No

Yes

Modeler_i

Modeler_i+1

Constraints_i+1

Model_i

Model_i
However, in the particular case of ArchiMate

The ArchiMate Business-Application alignment, from [OpenGroup, 2009]

The ArchiMate Application-Technology alignment, from [OpenGroup, 2009]
Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- State of the art: Advantages and limitations

Solution/Contribution:
1. Telecom service construction process
2. Tool building process
3. Software tools for telecom service construction

- Complete case study on a multimedia conferencing service
- Conclusion and perspectives
Case study: Modeling

1. Model
2. 
3. 
4. 

A multimedia conferencing service
Case study: Modeling with DSMLs a multimedia conferencing service

The model of a conferencing Service at the Telecom ArchiMate Business layer
Case study: Modeling with DSMLs a multimedia conferencing service

The model of a conferencing service at the Telecom ArchiMate Application layer
Case study: Modeling with DSMLs a multimedia conferencing service

The model of a conferencing service at the Telecom ArchiMate Technology layer
Case study: Testing

1. Modeler
2. Test
3. Describes the model from her viewpoint using her DSML
4. Interacts with the other Modelers

- Tests the model
  - No
  - Test satisfactory?
    - Yes

Integrate constraints from other views

Constraints

UDP Traffic Sent (Packets/Sec)
Statistic Sampling Period is 2 seconds.

Number of Packets
Case study: Testing through leverage of COTS a multimedia conferencing service

The static configuration of the conferencing service excerpt model for OPNET

OPNET simulation results for an IMS node of the conferencing service

OPNET=network simulator

http://www.opnet.com
Case study: Collaborating

1. [Diagram]
2. [Diagram]
3. Collaborate
4. [Diagram]
Case study: Collaborating by capturing Decision Rationale with a DR DSML on a multimedia conferencing service

Example of Collaboration Design Rationale ArchiMate extension used with a conferencing service example developed at the Application layer of the Telecommunications ArchiMate extension.
Agenda

- Challenges in telecom service life-cycle
- How to reduce telecom service construction time?
- State of the art: Advantages and limitations

Solution/Contribution:
1. Telecom service construction process
2. Tool building process
3. Software tools for telecom service construction

Complete case study on a multimedia conferencing service

Conclusion and perspectives
### Solution: Meeting the SCE requirements

<table>
<thead>
<tr>
<th>Req</th>
<th>An overall model</th>
<th>Domain specificity</th>
<th>Rapid prototyping</th>
<th>Collaborative support, Req 5</th>
<th>Early verification/simulation, Req 6</th>
<th>Integration</th>
<th>Reuse</th>
<th>Wide range of services</th>
<th>Easy evolution of services</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NGN**
- SIP (IMS): Cat 1, SIP - independent
- Cat 2: SIP - composition

**Cat 3: NGN**
- Parlay X

**Cat 4: Web**
- Web mash-up

**Cat 5: Hybrid**

**Solution**
- Related to the **individual** architecture of SCEs
- Not addressed by any of the categories
State of the art: Major limitations

- No category of SCE addresses **Req 1: An overall model**

- Choice between either:
  - **Req 2: Domain specificity** XOR
Solution/Contribution: Overcoming major limitations of the state of the art

No category of SCE addresses *Req 1: An overall model*

- Extending EAMLs (ArchiMate)

Choice between either:

- *Req 2: Domain specificity* XOR AND DSMLs
- *Req 3: Rapid prototyping, Req 7: Reuse, Req 8: Wide range and Req 9: Easy evolution*

- Models, code generation, MTs, HOTs, etc
Collaborative construction of telecommunications services

Solution/Contribution: advantages

- **Provides a method**
  Method = - a set of modeling conventions (Modeling Language - ML)
  - a process: - provides guidance as to the order of the activities,
  - specifies what artifacts should be developed using the ML.
  [Ramsin, 2008]

- **Reflects current practices in industry**
  (more easily) accepted by practitioners

- **Relies on models and offers high automation degree**
  more rapid software tool building

- **Contributes towards fulfilling all Service Providers' and Developers' requirements**
Solution/Contribution to the research question

How to reduce telecommunications service construction time while affecting non-negatively other parameters (e.g. Cost, QoS, QoE)?

RQ 1 Construction process

RQ 2 Software tools

RQ 3 Tool building process
Solution/Contribution: limitations

- The waterfall-like construction process is not flexible.
- Testing/simulation results have to be integrated manually into models.

- The SCE and construction process need testing by Service Providers and Developers.

- Viewpoint interoperability depends heavily on common entities.
Solution/Contribution: perspectives

- Introduce more flexibility (e.g., iterative, agile) in the construction process.
- (Semi-)automatic integrating of testing/simulation results into models.
- Construct Telecom DSML MMs from e.g. Frameworx
- Define alignment measures to ensure viewpoint interoperability (e.g. [Simonin, 2011])
- Implement the interoperability process


