Selecting Multimedia Service Compositions in Mobile Environments

Wolf-Tilo Balke and Jörg Diederich
L3S Research Center, Germany

ICWS’06
Chicago, IL, USA
Outline

- Motivation
- Multimedia Service Composition
  - Mobile Constraints
- Quality- and Cost-based Selection Model
- Use Case
- Summary and Future Work
Motivation

- Multimedia applications today are **monolithic**
  - Service-oriented framework introduce efficient reusability of components
  - Services introduce flexibility and best effort approaches
  - Service routing through network from media server to end user

Problem in Multimedia Services

- How to organize highly complex and dynamic workflows?
- How to exploit the deep understanding of multimedia data?

- **Web community** understands services – **Multimedia community** understands data
Motivation

- **Limited capabilities** in end user devices
  - Computationally complex or power-demanding tasks have to be moved to powerful servers
  - But also environment changes: movement of device,

- Basic Idea: **E²Mon Algorithm**
  - Monitors the execution chain of Web services
    - Graceful recovery from individual service failures
    - Graceful recovery from network-/device-specific alarms
  - Dynamically chooses the quality- and cost-optimal composition
    - Successive and parallel execution
Multimedia Service Composition

- Multimedia service composition is a composition **process**
  - Multiple services (e.g., retrieval, transcoding, display services) for processing and communication of multimedia data
  - Connected via functional and data dependencies to create a new multimedia service (e.g., a video-on-demand service)
  - Span over heterogeneous network and distributed system infrastructures

- Multimedia applications are usually **flow-based**
  - Data mostly continuous streams (e.g., video and audio streams)
  - Dependent in time and space
  - Stringent timing and spatial constraints on the functional services
  - Quality constraints (non-functional parameters) need to be taken into account
Multimedia Service Composition

- A **multimedia service** is a functional entity
  - With time, space and dependency relations to other services that precede or follow the application service
  - Functional dependency relations among individual multimedia services form a **service graph**
  - Service descriptions are expressed via meta-data and published in order for other services to be discovered and used

- **Mobile constraints**
  - Plethora of devices with different needs: data formats, ...
  - Very limited capabilities: display, computing power, ...
  - Power management: energy saving, battery low, ...
  - Transport difficult: low bandwidth, high costs, ...
Multimedia Service Composition

- Controlled by central instantiation and monitoring service
Some assumptions

- **No malicious services**
  - Capabilities are described correctly (MM-specific)
  - Provider discloses correct costs, QoS parameters, etc.

- **Service implementations for specific Web service types can be used **interchangeably**
  - Data is understood according to MPEG-7/21 descriptions

- **Central** control instance (monitor/proxy) caching results of previously executed Web services
  - Reuse in failure recovery
E²Mon Algorithm

- Run as a Web service on client or proxy machine
  - Proxy reduces communication load of mobile device for discovery, invocations, etc.
  - Efficient if client alarms are rare compared to external events (service failures,...)

- 4 basic phases
  - Workflow enumeration
  - Service discovery
  - Service chain selection
  - Execution monitoring
E²Mon Algorithm

- **Workflow enumeration**
  - Construct all possible workflows anticipated for multimedia applications (usage patterns)
  - **Input, output and intermediate/Transformational services**
    - Languages for workflow executions: BPEL4WS, ...
    - Languages for service capabilities: OWL-S, WSDL-S, ...

- **More complex enumeration**
  - AI planning techniques
  - State machines or Petri net-based approaches for simulation and verification
E²Mon Algorithm

- **Service discovery**
  - Service descriptions with multimedia metadata
  - Discover all suitable implementations
  - Classify due to constraints met/compromises needed
  - Extended UDDI, cooperative discovery

- **Periodic rediscovery**
  - Rediscovery on failure is too time-consuming (prefetching)
  - Execution parameters can change (e.g., QoS) more cost-effective solutions should be instantiated (hand-over)
  - Less important alarms (battery-low, etc.) can be handled, if better chains with respect to the alarm are available
E²Mon Algorithm

- **Service chain selection**
  - Calculate costs for all service chains
  - **Quantitative:**
    
    $F(I,t) = M(I) + Q(I) + w(t) \times P(I) + U(I)$
    
    - $M(I)$ – typical provider costs (service use, network use, etc.)
    - $Q(I)$ – quality of service costs (resolution, bandwidth, etc.)
    - $P(I)$ – power consumption costs, weighted by time variant function $w(t)$ dependent on device
    - $U(I)$ – user preference-based costs (decrease in MPEG-7/21 preference value, etc.)

- **Problem: very complex and sensitive functions**
E²Mon Algorithm

- **Preference-based** service chain selection
  - Get metadata given by MPEG-7/21 part 7 usage environment for digital item adaptation (DIA)
  - Metadata can include *transcoding hints*, *user interaction tools*, and *terminal capabilities*

- **Advanced digital item adaptation**
  - Handle complex preference trade-offs *qualitatively*
  - Still problematic in terms of efficiency
E²Mon Algorithm

- **Execution monitoring**
  - Execute services in a supervised fashion
  - Keep alternative chains updated (periodic rediscovery)
  - Handle service failures by reasessing cost for the same chain with the next best implementation of the failed service
  - Change to different implementation or different chain

- **Possible alarms**
  - Service failure needs graceful recovery
  - Change in service parameters, especially QoS
  - Another more cost-effective chain becomes executable
  - Local events (roaming to new network, battery warning, user interaction, ...)
Media Streaming Use Case

- FIFA Soccer WorldCup 2006
  - All games streamed using digital video broadcasting standard (DVB-T / DVB-H)
  - Moving through networks with a mobile device (PDA)
  - Small screen, network technologies: WLAN, UMTS, GPRS
Media Streaming Use Case

- Enumeration of possible service chains
  - Planning is not the major problem of multimedia applications
  - Mixture of **content** and **transport** services
  - Hard constraints might **rule out** some of the compositions

![Diagram of service chains]
Media Streaming Use Case

- Sequential and parallel execution, e.g. video transportation service T1
  - Bundles different transportation technologies, transcoding, etc.
  - But also: billing services, etc.
Media Streaming Use Case

- **Possible service chains**
  - Service descriptions enable determining executable sequences
  - Different implementations/providers may exist
  - Services can interpret data by exploiting metadata (MPEG-7/21)

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Service chain</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S1-S2-S4-S5</td>
<td>Video-WLAN</td>
</tr>
<tr>
<td>2</td>
<td>S1-S2-S4-S6</td>
<td>Video-UMTS-hiQ</td>
</tr>
<tr>
<td>3</td>
<td>S1-S2-S4-S7-S8</td>
<td>Video-UMTS-lowBw</td>
</tr>
<tr>
<td>4</td>
<td>S1-S2-S4-S7-S9</td>
<td>Video-GPRS-lowBw</td>
</tr>
<tr>
<td>5</td>
<td>S1-S2-S10-S9</td>
<td>Mobile Video-GPRS</td>
</tr>
<tr>
<td>6</td>
<td>S1-S3-S11-S5</td>
<td>Newsticker-WLAN</td>
</tr>
<tr>
<td>7</td>
<td>S1-S3-S11-S9</td>
<td>Newsticker-GPRS</td>
</tr>
</tbody>
</table>
Media Streaming Use Case

- Compare possible chains and implementations

<table>
<thead>
<tr>
<th>Nr.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>S_exec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S1</td>
<td>S2</td>
<td>S4</td>
<td>S5</td>
<td>nil</td>
<td></td>
<td>True</td>
</tr>
<tr>
<td>2</td>
<td>S1</td>
<td>S2</td>
<td>S4</td>
<td>S6</td>
<td>nil</td>
<td></td>
<td>True</td>
</tr>
<tr>
<td>3</td>
<td>S1</td>
<td>S2</td>
<td>S4</td>
<td>S7</td>
<td>S8</td>
<td>nil</td>
<td>True</td>
</tr>
<tr>
<td>4</td>
<td>S1</td>
<td>S2</td>
<td>S4</td>
<td>S7</td>
<td>S9</td>
<td>nil</td>
<td>True</td>
</tr>
<tr>
<td>5</td>
<td>S1</td>
<td>S2</td>
<td>S10</td>
<td>S9</td>
<td>nil</td>
<td></td>
<td>False</td>
</tr>
<tr>
<td>6</td>
<td>S1</td>
<td>S3</td>
<td>S11</td>
<td>S5</td>
<td>nil</td>
<td></td>
<td>True</td>
</tr>
<tr>
<td>7</td>
<td>S1</td>
<td>S3</td>
<td>S11</td>
<td>S9</td>
<td>nil</td>
<td></td>
<td>True</td>
</tr>
</tbody>
</table>

Service index

- Service chains

Executebility information
Media Streaming Use Case

- Cost-table (w/o preference values)
- Based on implementations

<table>
<thead>
<tr>
<th>Implementation</th>
<th>M</th>
<th>Q</th>
<th>P</th>
<th>U</th>
<th>F(I, t) w(t) = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I4a</td>
<td>0</td>
<td>0</td>
<td>64</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>I4b</td>
<td>1</td>
<td>0</td>
<td>64</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>I5a</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>I5b</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>I6</td>
<td>15</td>
<td>1</td>
<td>8</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>I7a</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>I7b</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>I8</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>I9</td>
<td>3</td>
<td>15</td>
<td>4</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>I11</td>
<td>1</td>
<td>17</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
</tbody>
</table>
Media Streaming Use Case

- Calculate \( F(I, t) \) for all chains

<table>
<thead>
<tr>
<th>Nr.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>( V_{\text{min}} )</th>
<th>( V_{\text{chain}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I1</td>
<td>I2</td>
<td>I4b</td>
<td>I5a</td>
<td>nil</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>I1</td>
<td>I2</td>
<td>I4b</td>
<td>I6</td>
<td>nil</td>
<td>26</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>I1</td>
<td>I2</td>
<td>I4b</td>
<td>I7b</td>
<td>I8</td>
<td>nil</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>I1</td>
<td>I2</td>
<td>I4b</td>
<td>I7b</td>
<td>I9</td>
<td>nil</td>
<td>26</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>I1</td>
<td>I3</td>
<td>I11</td>
<td>I5a</td>
<td>nil</td>
<td>23</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>I1</td>
<td>I3</td>
<td>I11</td>
<td>I9</td>
<td>nil</td>
<td>37</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Minimal cost chain is instantiated
Media Streaming Use Case

- Monitoring dynamics
  - Service chains change executability status, e.g., joining or leaving a network (WLAN, GPRS)
  - Discovery of new service implementations, e.g., mobile video service
  - Local events, e.g. low battery changes best possible service from streaming to newsticker (preference vs. energy)
Summary

Problem

- Multimedia Services demand deep understanding of data and flow-based compositions
- Mobile devices have limited capabilities and pose additional constraints
- Environmental parameters may change quickly

Idea

- E²Mon algorithm selects the most cost-efficient chain for instantiation
- Continuously monitors the execution and handles alarms
- Dynamically controls costs based on providers, quality of service, power consumption and user preferences
Future Work

- Further investigation of E²Mon’s applicability
  - Currently a **large testbed** is built to investigate scalability constraints
  - Find good application-dependent **defaults** for cost functions and parameters (e.g., optimal rediscovery time)
  - Consider **qualitative** functions for service selection and introduce new more expressive preference model
Questions?
balke@l3s.de