Criteria for the Evaluation of Self-* Systems

Elsy Kaddoum¹, Claudia Raibulet, Jean-Pierre George¹, Gauthier Picard², Marie-Pierre Gleizes¹

¹IRIT – University Paul Sabatier, Toulouse, France
²Ecole Nationale Superieure des Mines, Saint Etienne, France

Università degli Studi di Milano-Bicocca

DISCo – Dipartimento di Informatica Sistemistica e Comunicazione

Milan, Italy

SEAMS 2010
Software Engineering for Adaptive and Self-Managing Systems
Cape Town, South Africa
Adaptivity Issues

- The development of adaptive systems is not a trivial task due to:
  - a conceptual reason:
    - how can be adaptivity defined?
    - what is actually an adaptive system?
    - which are the main features a system should meet in order to be considered adaptive?
  - a practical reason:
    - what should be adapted/changed in the structure and/or behavior of a system at runtime?
    - how should be adapted/changed in the structure and/or behavior of a system at runtime?
Proposal

- Identify and define a set of software criteria which capture the essentials of adaptivity and based on which adaptive systems, applications, and frameworks for the development of adaptive systems, may be evaluated in order to be compared or integrated or ...
Our Approach

- Studied the available approaches to address adaptivity from various points of view:
  - design
  - advantages and limitations
  - performances
  - evaluation provided by authors
  - ...

Essentially, which are the main aspects related to adaptivity described by the authors of the adaptive system and of the frameworks for the development of adaptive systems
Definitions of Adaptivity

- **Architectural** adaptation
  - changes made **at run-time** in the structure of the components of a system and/or in the interactions among them by using an architectural model of a system [Garlan et al., 2004]

- **Compositional** adaptation
  - modifications of the structure and behavior of a software made **at run-time** due to the changes occurred in its execution environment [McKinley et al., 2004]

- **Structural** adaptation
  - changes of the type of application components made **dynamically** while preserving its behavior and services [Gorton et al., 2007]

- **Behavioral** adaptation
  - changes made **dynamically** in the execution of the software components in a non intrusive way (e.g., by changing its configuration) [Gorton et al., 2007]

- **Content** adaptation
  - transformation and manipulation of contents **at run-time** based on the features of the application or device requiring them [He et al., 2007]

- **Service** adaptation
  - translated into content as well as behavioral adaptation [Choi et al., 2007].
Adaptivity

- Indicates the capability of a system
  - to perform changes by itself
  - at runtime
  - due to changes occurred internally or externally

- Changes have various objectives:
  - automation of human tasks
  - performances and productivity improvement
  - enhancement of functionalities
  - reduction of complexity
  - management of critical situations
  - ...

- Adaptivity - a complex issue to address, and it is even harder to evaluate

- Changes regard various aspects at various levels:
  - architectural
  - structural
  - functional
  - content
  - service
  - ...

Available Evaluation Approaches

- Solutions specific to an application domain
- Solutions considering one particular perspective (e.g., user-oriented)
- Metrics for software adaptability [Subramanian, Chung, 1999] - three indices:
  - **Element Adaptability Index (EAI)**
    - 0, for non-adaptable elements
    - 1, for adaptable elements
  - **Architecture Adaptability Index (AAI)**
    - EAI for all the elements of an architecture / Total number of elements
  - **Software Adaptability Index**
    - AAI for all the architectures of a software / Total number of architectures for that software
Open Issues

- Various solutions to address and implement adaptive issues
- Different perspectives on adaptivity:
  - methodological
  - architectural
  - intrinsic
  - performance
  - ...

- What should it be evaluated?
- How should it be evaluated?
- How can systems be compared against their adaptivity?
- Is it possible to determine the degree of adaptivity of a system?
Criteria for the Evaluation of Adaptivity

- **Four categories:**
  - methodological
  - architectural
  - intrinsic
  - runtime
Methodological Criteria

- **Genericity Index**
  - Indicates the effort needed to adapt an approach to a given problem

- **Adaptivity Distribution Index**
  - Indicates the distribution of the adaptive elements on the physical nodes of an adaptive system
Architectural Criteria

- **Separation of concerns**
  - Indicates the dependence between the functional part of a system and its adaptive part

- **Adaptivity Pattern Index**
  - Indicates the separation of concerns among the four steps of adaptivity
Architectural Criteria

- **Minimum Adaptivity Growth**
  - Indicates the minimum number of elements required to make a system adaptive

- **Adaptivity Growth per Functionality**
  - Indicates the number of elements needed to introduce the i-th adaptivity feature

- **Overall Adaptivity Growth**
  - Represents the sum of the previous two criteria
Intrinsic Criteria

- **Domain Factors Influencing Adaptivity**
  - Indicates the number of domain specific factors taken in input by the adaptive logic

- **Systems Factors Influencing Adaptivity**
  - Indicates the number of system specific factors taken in input by the adaptive logic
Intrinsic Criteria

- **Local Computational Complexity**
  - Analyses the local adaptation algorithms of an agent

- **Decentralization Degree**
  - Indicates how the decision process is distributed among the agents
Intrinsic Criteria

- **Automated Administration Tasks**
  - Indicates the administration tasks automated through adaptivity

- **Average User Interaction per Functionality**
  - Indicates the average number of interactions users perform with the system to require a functionality
Performance Criteria

- **Latency**
  - Indicates the variation of the response time in the presence of adaptivity with respect to the response time when no adaptivity mechanisms are used.

- **Working vs Adaptivity Time**
  - Indicates the time needed to provide a functionality vs the time needed for adaptivity.

- **Communication Load**
  - Indicates the time needed for communication in a decentralized solution.
Performance Criteria

- **Non Determinism**
  - Indicates the difference between solutions when the same adaptivity issue is addressed

- **Storage Dimension Growth**
  - Indicates the physical storage growth due to the presence of the adaptive mechanisms

- **Quality of Response**
  - Indicates the variation of the quality of response in the presence and absence of adaptivity
Performance Criteria

- **Robustness Index**
  - Indicates the stability of an adaptive system in the presence of adaptation needs

- **Time for Adaptation**
  - Indicates the time needed by the system to achieve a normal functioning after an adaptation need has occurred
Application of Criteria

- Methodological and architectural criteria
  - quality of the design of adaptive systems

- Intrinsic and performance metrics
  - quality of the functionalities of adaptive systems
<table>
<thead>
<tr>
<th>Case Study</th>
<th>Goal of Adaptivity</th>
<th>Adaptivity Type</th>
<th>Main Steps of Adaptivity</th>
<th>Models</th>
<th>Applicable Categories of Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web-based Client Server</td>
<td>Automate administrative tasks</td>
<td>Architectural</td>
<td>Number of servers</td>
<td>Mapping / reflecting the changes at the functional level</td>
<td>Rainbow – definition of specific architectural styles</td>
</tr>
<tr>
<td></td>
<td>Performance optimization</td>
<td></td>
<td>Bandwidth Response time</td>
<td></td>
<td>Architectural: SCI, API, MAG Intrinsic: DFIA, SFIA, FIA, LCC, AAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Runtime: Latency, WAT, RI, TA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Constraints evaluation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>through invariants</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Adaptive strategy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mapping / reflecting the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>changes at the functional</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>level</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rainbow - definition of specific architectural styles</td>
<td>Methodological: GI Architectural: SCI, API, MAG Intrinsic: DFIA, SFIA,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FIA, LCC, AAT Runtime: Latency, WAT, RI, TA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video-conference</td>
<td>Performanc e optimization</td>
<td>Architectural</td>
<td>Bandwidth</td>
<td>Resource allocation (placement of containers, stations, and operators)</td>
<td>The AMAS model using the ADELFE methodology and the JavAct middleware</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Gateway cost</td>
<td></td>
<td>Methodological: AAII, ADI, GI Architectural: MAG Intrinsic: LCC, DD,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ALI, ANI, AAT, AvgUIF Runtime: Latency, WAT, CL, QoR, ND, Progress,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SDA, RI, TA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Constraints evaluation</td>
<td>Cooperati ve Strategy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>through invariants</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAMAS COP</td>
<td>Performanc e optimization</td>
<td>Behavioral</td>
<td>Containers, Stations</td>
<td>Resource allocation (placement of containers, stations, and operators)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Operators (The factory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Schedule)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Constraints evaluation</td>
<td>Cooperati ve Strategy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>through invariants</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Adaptive strategy</td>
<td>Resource allocation (placement of containers, stations, and operators)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mapping / reflecting the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>changes at the functional</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptive Image Server</td>
<td>Performanc e optimization</td>
<td>Behavioral</td>
<td>Image resolution</td>
<td>Modify image</td>
<td>Adaptive Server Framework – personalizatio n of the framework</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bandwidth</td>
<td></td>
<td>components</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU load</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Throughput</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Methodological: AAII, GI Architectural: SCI, API, MAG, AGF, OAG, AvgAG,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GAE Intrinsic: DFIA, SFIA, FIA, LCC, AAT, AvgUIF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Runtime: Latency, WAT, QoR, ND, RAMP, CPUP, SDG, RI, TA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AHA!</td>
<td>Enhance functionalities</td>
<td>Content</td>
<td>Rule based users’ update</td>
<td>Modify content Modify navigation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Navigation</td>
<td>profiles</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>User knowledge</td>
<td>Rule based adaptation of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>level</td>
<td>content</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rule based adaptation of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>navigation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Methodological: AAII Intrinsic: SCI, API, MAG, AGF, OAG, AvgAG, GAE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DFIA, SFIA, FIA, LCC, AAT, AvgUIF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Runtime: Latency, WAT, QoR, ND, RAMP, CPUP, SDG, RI, TA</td>
</tr>
</tbody>
</table>
Comparing Adaptive Systems
Conclusions

- Metrics for the evaluation of adaptivity provide:
  - A common vocabulary for the development of adaptivity
  - A common and unified mechanism for the evaluation of the quality of the design and the performances of adaptive properties
  - A first step towards:
    ⇒ the analysis of adaptive systems for their quality
    ⇒ the comparison of adaptive approaches for their interoperability and integration
Further Work

- Extension of the defined criteria
- Application of the criteria to more case studies to determine statistically their usability and usefulness
- Application of the criteria during the engineering of adaptive systems (e.g., when non-functional requirements are identified, specified, implemented)
- Definition of a benchmark for adaptivity in information systems