Towards a Model for Quality of Web and Grid Services

Sergio Andreozzi (speaker)
Paolo Ciancarini
Danilo Montesi
Rocco Moretti
OUTLINE

- Quality of Service as Satisfaction Measurement
- Background on
  - Measurement Theory
  - Logic Scoring of Preferences
- Proposal for a Service Quality Model
- Example: a Grid Computing use case
Service Oriented Architecture

type of distributed systems where participating agents act as service requesters and service providers exchanging messages through a shared transport mechanism

- **Service**: an abstracted, logical view of actual programs, databases, business processes, etc., defined in terms of what it does, typically carrying out a business-level operation

- Two main incarnations: WSA and OGSA
Evaluation and Selection of a Service

- In SOAs, a variety of services can be available to be automatically discovered and selected for subsequent usage.
- The selection is based on Service Requestor’s expectations.
- Expectations
  - are expressed against a set of service characteristics
  - can be numerous
  - can show inter-dependencies
  - can be partially satisfied (i.e., the requestor’s satisfaction is typically a ‘continuous function’)
  - can be categorized in relevance classes
An example in the Grid context: the Actors

- **Computing Service (CS):** a uniquely identified Grid service providing a user software application for computing power in a certain execution environment

- **Grid User:** expresses its expectations in terms of constraints and a ranking expressions as a Job Submission Description (JSD) document

- **Resource Broker:** Grid service that receives a JSD and finds the best match
An example in the Grid context

- A widely used language for expressing capabilities and expectations is the Classified Advertisement (ClassAd) language.

- It enables a matchmaking process among a set of available services and a set of service requests.
An example in the context of Grid: ClassAds

[  Type = "CS";  
  URI = "serviceURI1";  
  OSType = "RedHat Linux";  
  MaxWallClockDuration = 2000;  
  TotalCPUConsumption = 1000;  
  FreeJobSlots = 56;  
  RunningJobs = 25;  
  WaitingJobs = 0  
]

[  Type = "JSD";  
  Executable = "myApplication";  
  Arguments = "-arg1 myArg1";  
  FileStageIn = {"fileURI1"};  
  FileStageOut = {"fileURI2"};  
  Constraint =  
      other.Type == "CS" &&  
      other.OS_Type == "RedHat Linux";  
  Rank = other.FreeJobSlots;  
]
Quality of Service

- **Quality of Service** is expressed in terms of the satisfaction degree expected by the requestor.
- We propose a model for a quantitative evaluation of such a satisfaction degree.
- This model relies on measurement theory and the Logic Scoring of Preferences (LSP) method.
Measurement Theory

**Measurement**: an activity allowing analysts to describe by numbers the attributes of the entities under examination

- It enables a rigorous representation of the attributes involved in the quality evaluation
- It helps in focusing on the entities, relationships and operations of interest and their mapping to a rigorous form suitable for machine processing
Measurement Theory

- **Measurement Scale**: for each attribute, a triple comprising (1.) values, relationships and operations ‘as perceived by humans’, (2.) values, relationships and operations ‘abstracted in a rigorous model’, and (3.) a measure function providing the mapping from 1. to 2. preserving relationships and operations properties among values.

- Important measurement scales categories are:
  - **Nominal scale**: classification of measurement values
  - **Ordinal scale**: order of measurement values
  - **Interval scale**: interval between measurement values
  - **Ratio scale**: ratio between measurement values
  - **Absolute scale**: single possible measurement
Logic Scoring of Preferences

Logic Scoring of Preferences (LSP):

a quantitative method for evaluating, comparing and selecting hardware and software systems

It consists of two steps:

1. **Elementary Criteria of Satisfaction**: definition of a function for each attribute in order to quantify the satisfaction associated to its possible values
   - A number $e \in [0, 1]$ is associated to each value
   - From $e=0$ (no satisfaction) to $e=1$ (full satisfaction) there are intermediate satisfactions
Logic Scoring of Preferences

2. **Aggregation**: the global satisfaction $E$ for $n$ satisfactions $e_1, e_2, \ldots, e_n$ is obtained by applying the functions:

$$E = (w_1 e_1^r + \ldots + w_n e_n^r)^{1/r}$$

where: $-\infty \leq r \leq +\infty$; $w_i \geq 0 \ \forall \ i \in [0, 1]$; $\sum_{i=0}^{n} w_i = 1$

- Satisfactions can be partitioned in categories of relevance (i.e., essential, desirable, and optional) and then aggregated
- The power $r$ captures the desired logical relationship and intensity of polarization (e.g., it enables to express replaceability or simultaneity of inputs)
Logic Scoring of Preferences

the general aggregation pattern

WETICE-2004 Conference - 14 Jun '04
Modena, Italy
Our proposal

- Service Quality Model:
  - \( AM=\{(a, S_a) | a \text{ is an attribute and } S_a \text{ is a measur. scale}\} \)
  - \( \sim REL=(a_i, a_j) \) is an equivalent relation concerning relevance of attributes
  - \( EC_A \) is the set of elementary criterion of satisfactions for each attribute in \( A \)
  - \( AF_{EC_A} \) is the set of aggregation functions for the synthesis of the overall score
Example: (cont.) ClassAds

[  
  Type = "CS";
  URI = "serviceURI1";
  OSType = "RedHat Linux";
  MaxWallClockDuration = 2000;
  TotalCPUConsumption = 1000;
  FreeJobSlots = 56;
  RunningJobs = 25;
  WaitingJobs = 0
]

[  
  Type = "JSD";
  Executable = "myApplication";
  Arguments = "-arg1 myArg1";
  FileStageIn = {"fileURI1"};
  FileStageOut = {"fileURI2"};
  Constraint =
    other.Type == "CS" &&
    other.OSType == "RedHatLinux";
  Rank = other.FreeJobSlots;
]
Example: OSType attribute

Nominal Scale

RedHat Linux
Debian Linux
WinXP

µ  µ  µ

1  0  2

Elementary Criteria of Satisfaction for OSType
Example: FreeJobSlots attribute

\[ \mu = |FJS| \]

Ordinal Scale

\( N \)

Elementary Criteria of Satisfaction for FreeJobSlots

\[ i. \quad FJS_{\text{MAX}} = 20 \]

WETICE-2004 Conference - 14 Jun '04
Modena, Italy
Example: overall satisfaction

- Parameters can be selected in order to enforce OSType to be an essential attribute and FreeJobSlots affecting the OSType satisfaction for a certain value range.
Conclusions

- SOAs require automatic evaluation and selection capabilities of services in respect of a set of expectations.
- The proposed model offers a rigorous method for attribute representation and a quantitative approach for the measure of the overall satisfaction degree.
- Ongoing work is focusing on:
  - Composite service selection use case
  - Meta-model definition based on OMG MDA standard
  - Query language extension for expressing users' expectations in terms of satisfaction