# Context-aware Adaptation Based on Distributed Policies: an Experimental Service

## Demo Storyboard

Claudio Bettini, Daniele Riboni
DICo, University of Milan


## Abstract

The heterogeneity of device capabilities, network conditions and user contexts that is associated with mobile and ubiquitous computing has emphasized the need for effective adaptation of internet services. The architecture that we are developing at the University of Milan has the objective of managing distributed profile information and adaptation policies, solving possible conflicts by means of an inference engine and prioritization techniques. We plan to conduct the demo of our framework by showing the behavior of various components of the architecture in the context of a web based application. The structure of the demo follows.


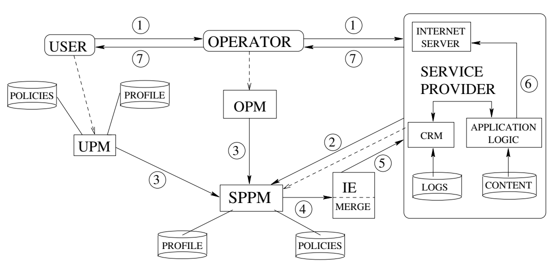


*Figure 1: Architecture*

## Introduction

After a brief introduction to the global architecture depicted in Figure 1, we will explain the profile aggregation and policy evaluation mechanisms implemented by our system. Examples of partial profiles storing different values for a single profile attribute

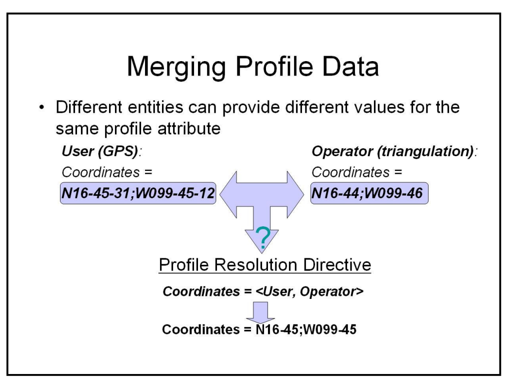


*Figure 2: Profile Aggregation*

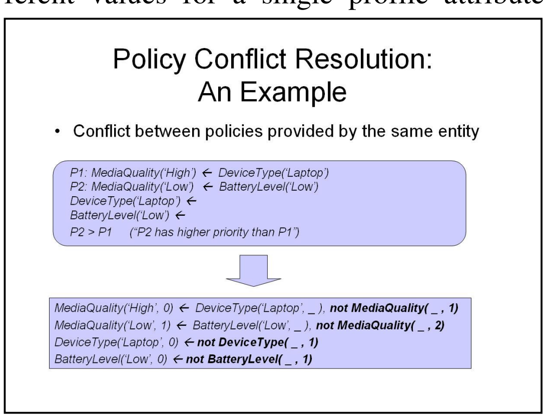


*Figure 3: Policy Conflict Resolution*

and the profile resolution directives used to handle these kind of conflicts will be illustrated with the help of slides like the one in Figure 2. That slide represents the following situation: The User profile provides a value *X* for the *Coordinates* attribute, while the Network Operator provides a different value *Y* for the same attribute. Profile aggregation is performed by the Merge module that resolves the above conflict by applying the profile resolution directive ***Coordinates=<User, Operator>***. Thus, in the aggregated profile the value of the *Coordinates* attribute is set to *X*.

In order to illustrate the mechanism of conflict resolution between policies, a simple example of a conflict between two rules declared by the same entity over the same attribute will be shown using slides. In Figure 3, a conflict is shown between two policies (P1 and P2) declared by the Service Provider which set a value to the *MediaQuality* attribute. The Service Provider gives higher importance to the policy P2. In order to handle the conflict, the Inference Engine Preprocessor modifies the input program as shown in the bottom of the slide. A second parameter is added to predicates, in order to include the *weight* of the rule. Each rule with a weight *w* that sets a value for the *P* attribute (i.e., whose head predicate is *P*) can fire only if  the rule that sets a value to *P* with weight *w+1* cannot. Note that each rule with weight *w* can be evaluated only after the evaluation of the rule with weight *w+1* that sets a value to the same attribute (if such a rule exists). This feature is garanteed by an appropriate evaluation strategy.

## *Profiles and policies at work: the ProtoTown web application*

The prototype web application we developed is a micro-portal addressed to tourists visiting the fictitious *ProtoTown*. The portal shows an ordered list of Points Of Interest (POIs), i.e. hyperlinks that point to a web page providing multimedia information of a physical place (e.g., museum, restaurant), as shown in Figure 4.


*Figure 4: a screenshot of the application*

The application logic maintains a database which assigns a location and an interest category to each POI. The delivery of content is performed by Cocoon, which uses this information to select and sort POIs according to the user's  interests and his distance from the POI.

The multimedia content to be delivered (see Figure 5) is further selected by taking into account profile data such as the user device capabilities, the network conditions as well as user and service provider policies.


*Figure 5: multimedia content*

In order to show some of the priority rules and policies which determine the service adaptation, and how they affect the selection of the content, and its presentation, we will consider the case of John, a business man travelling to ProtoTown.

John is browsing using his PDA while moving across the city. The micro-portal continuosly adapts to the changes of context

(location, network conditions, time of the day) of John, showing different POIs and multimedia content.
These changes are driven by the IE upon the evaluation of policies declared by both the user and the service provider; policies generate different directives based on the different integrated profile obtained from the Merge module.

## Profile Resolution Directives

Priority rules over attributes are declared by the service provider in order to resolve conflicts due to different values provided by different entities for the same attribute. Profile resolution directives are stored in an XML file (see Figure 6) and can be modified through a proper web based interface. Some of the profile resolution directives declared by the ProtoTown service provider are given below:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <PriorityRules>
  - <Rule>
    - <Target>
        <Component>*.*</Component>
        <Attribute>*.*</Attribute>
      </Target>
      <Value>SPPM,UPM,OPM</Value>
    </Rule>
  - <Rule>
    - <Target>
        <Component>DemoVoc.Interests</Component>
        <Attribute>DemoVoc.InterestCategories</Attribute>
      </Target>
      <Value>UPM,SPPM</Value>
    </Rule>
```

*Figure 6: profile resolution directives*

(1) setPriority InterestCategories = (UPM, SPPM)
(2) setPriority Coordinates = (UPM, OPM)
(3) setPriority MediaQuality = (SPPM, UPM)

## User and service provider policies

Policies are stored as RuleML files in proper repositories of the User and Service Provider Profile Managers (Figure 7 shows the RuleML file of the user's policies). Policies are declared by the user in order to express his preferences regarding the content and its presentation. John declared the following policy to request high-quality multimedia content when using his PDA:

(4) If  DeviceType = 'Pda'  Then Set  MediaQuality = 'High'.

Similarly, service providers can declare policies for determining content and presentation directives.
The following policy, declared by the ProtoTown service provider, states to deliver low-quality multimedia contents when the available bandwidth drops below a certain threshold:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <rulebase>
  - <imp>
    - <_head>
      - <atom>
        - <_opr>
            <rel>Phase</rel>
          </_opr>
          <ind>noon</ind>
        </atom>
      </_head>
    - <_body>
      - <and>
        - <atom>
          - <_opr>
              <rel>CurrentTime</rel>
            </_opr>
            <var>X</var>
          </atom>
        - <atom>
          - <_opr>
              <rel>GREATER</rel>
            </_opr>
            <var>X</var>
            <ind>11</ind>
```

*Figure 7 : user policies*

(5) If  AvailableBandwidth  <  56kbps  Then Set  MediaQuality = 'Low'.

Policies can also be declared in order to enrich the profile.
The following service provider policy induces the phase of the day analyzing the current time:

(6) If UserCurrentTime > 11:30 And UserCurrentTime < 13:30 Then Set Phase = 'Noon'.

In order to handle possible conflicts between different rules which could set different values to the same attribute, policies are modified before evaluation, as shown in Figure 8.
In particular, policy 5 has a weight higher than the one of policy 4, since the entity that declared it has a higher priority in the profile resolution directive (3) relative to the *MediaQuality* attribute.

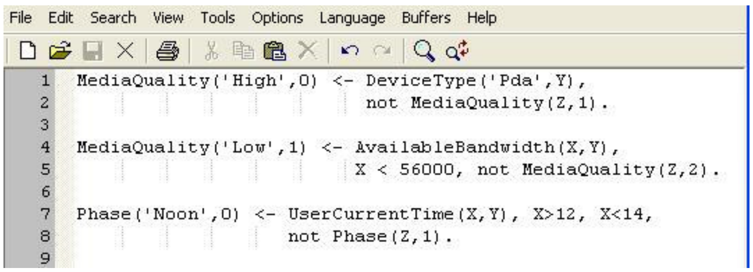


*Figure 8: the logic program after preprocessing*

## Scenario 1: Arriving at ProtoTown

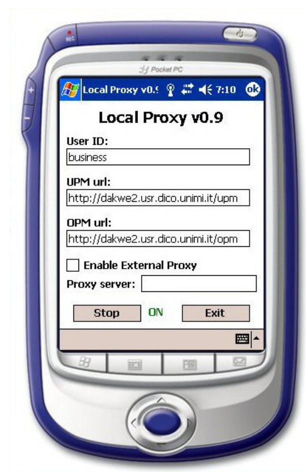


*Figure 9: the local proxy*

John arrives at ProtoTown by train at 9 AM. Since he's never been there, he searches for information about the city browsing the ProtoTown portal.
At first, he activates the local proxy that will attach profile references to HTTP request headers (see Figure 9). When he connects to the portal, the list of POIs is shown.
The Merge module obtains two different values for the Coordinates attribute: one is provided by John's GPS device, one is provided by the network operator on the basis of the user's current cell.
Applying the resolution directive (2), the value chosen by the Merge is the one provided by the user's device.
Following the directive (1), the list of interests is ordered by considering the interests declared by John followed by the ones induced by the service provider.
Intra-category POIs are ordered by their proximity to the user. The list of POIs is shown in Figure 10.
Since John is browsing via his PDA, the evaluation of policy (4) sets the attribute *MediaQuality* to "High"; thus, when John selects the link "Baseball Stadium", he re-



*Figure 11: high resolution content*

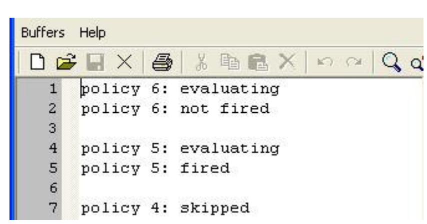


*Figure 12: IE logs*

ceives high quality multimedia content (see Figure 11). Note that the policy (5) cannot fire, since we suppose John is currently in a WiFi hot-spot. The Inference Engine logs (shown in Figure 12) show the order of evaluation of rules. We can see that policy 5 is evaluated first; since it does not fire, policy 4 is evaluated next, and sets a value for the *MediaQuality* attribute.

### Scenario 2: At noon

At 12 AM, John is downtown. The list of POIs he obtains is different: restaurants appear on top (see Figure 13.). This is due to the firing of the service provider policy (6), that sets the value of the Phase attribute to "Noon"; the application logic then adds the restaurants category on top of the user's interests (see the Inference Engine logs in Figure 14).

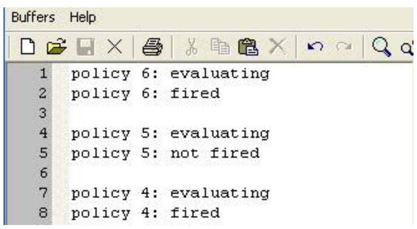


*Figure 14 : IE logs*

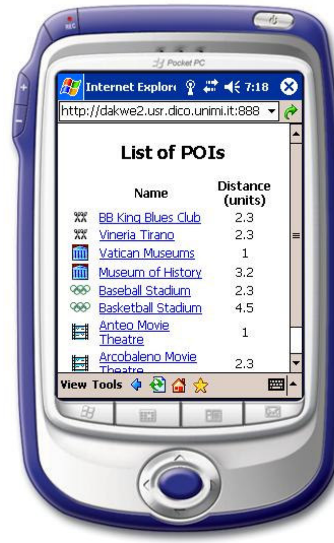


*Figure 13: the list of POIs at noon*

Moreover, John is still browsing via his PDA, but is no more covered by a WiFi network. For this reason, both policy (4) and policy (5) could fire, deriving incompatible values for the MediaQuality attribute. Once more, policy 5 is evaluated first, since his weight is higher than the one of policy 4. Since policy 5 fires, policy 4 is not evaluated, and the attribute MediaQuality is set to "Low".
As a consequence, when John selects the link "Anteo Movie Theater", he receives low quality multimedia content for available trailers (see Figure 15.).


*Figure 15: low quality content*

## *Performance Evaluation*

If demo attendants are interested we will go into some details about system performance and scalability.
Given the simple nature of the prototype scenario, adaptation was performed considering a small number of rules, and essentially no appreciable delay due to conflict resolution can be observed in response time. In order to estimate the feasibility of the evaluation of policies

based on logic programming for more sophisticated services, we performed an experiment using artificial rulesets of various cardinalities. The rulesets are evaluated by a well-known inference engine (the *DLV* solver, http://www.dbai.tuwien.ac.at/proj/dlv/) and by the ad-hoc inference engine we implemented.

The rulesets are built as follows: Each rule in the rulesets has three random subgoals, one of which is negative; for each attribute, each ruleset contains three conflicting rules.

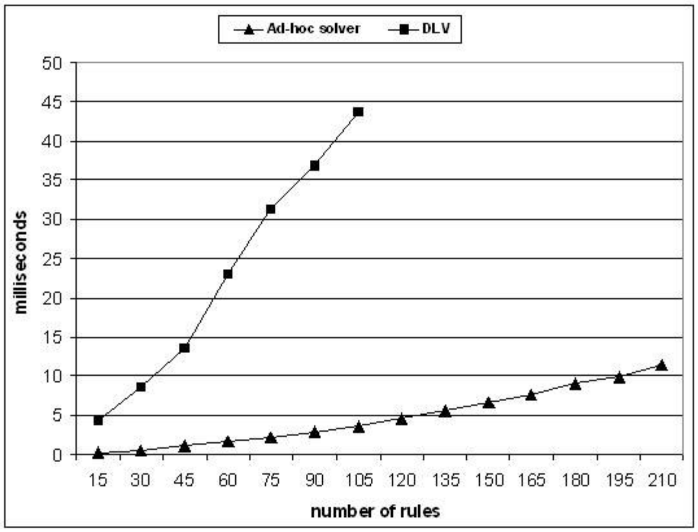


*Figure 16: Results with DLV and the ad-hoc solver*

In order to evaluate the rulesets with *DLV*, rulesets must be acyclic. Moreover, syntax has been slightly modified in order to be acceptable by *DLV*.

Figure 16 shows experimental results of executing the rulesets on a two-processor Xeon 2.4 GHz workstation.

Evaluation times are averages of ten runs, each using a different random ruleset: Results with DLV show that a ruleset of 15 rules is evaluated in around 5 milliseconds, while a ruleset of 105 rules is evaluated in around 40 milliseconds.

The same rulesets have been evaluated by the ad-hoc inference engine we developed; Figure 16 shows that performance is remarkably improved.

It should be noted that evaluation times with both engines exhibit a linear increase with the number of rules in the ruleset. Considering that, for most internet services, adaptation will probably be performed considering a small subset of the user and the service provider policies, rulesets evaluation time seems to be acceptable for this class of services; Therefore, we expect that response time will be dominated by the network latency.