UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES

Laboratoire d'informatique CNRS - UMR 8144 PRiSM

Médiation Ambiante : sélection dynamique de ressources

Kim-Tam HYUNH* Béatrice FINANCE*

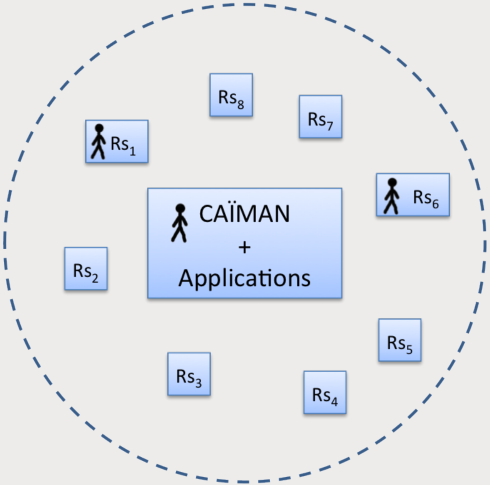
* PRiSM / CNRS

Motivations

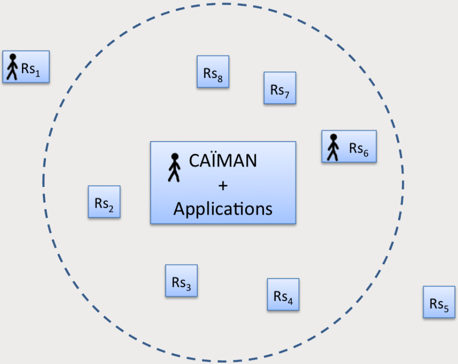
Towards a Context-aware dAta Integration and Management for Ami eNvironnments that :

- Facilitates personalized and contextual integration and monitoring of heterogeneous data streams through continuous queries execution.
- Enables applications to dynamically sense and control, according to some preferences the ambient environment of the user, which is changing over time and space.
- Enables the user to keep some control over his personal data as the monitoring is done exclusively on his personal device.

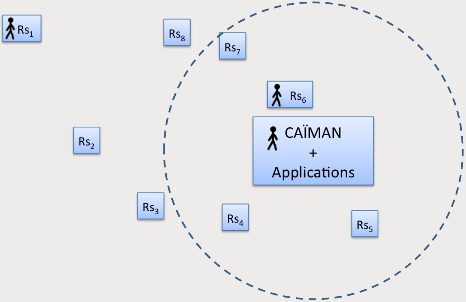
Mobility in open and pervasive Environment



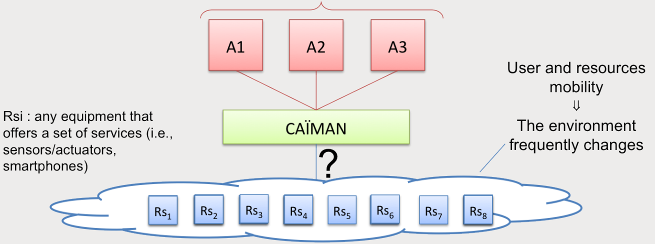
Mobility in open and pervasive Environment



Mobility in open and pervasive Environment

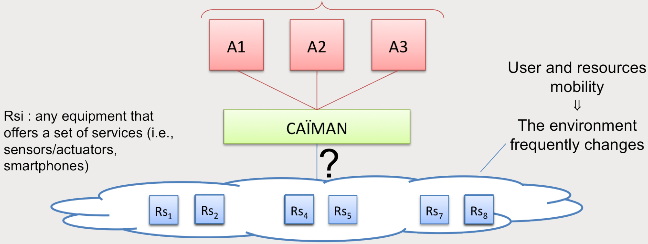


Dynamicity of the Ambient Environment



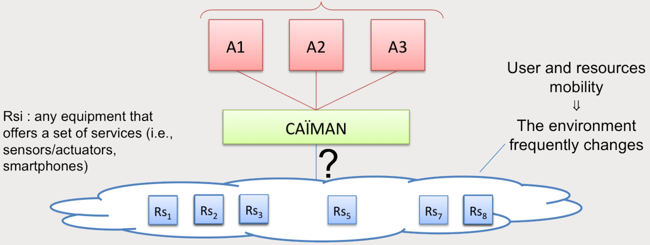
→ What are the most relevant resources in this dynamic context ?

Dynamicity of the Ambient Environment



→ What are the most relevant resources in this dynamic context ?

Dynamicity of the Ambient Environment



→ What are the most relevant resources in this dynamic context ?

Towards an Ambient Data Mediation System (1)

The following requirements clearly distinguish an ambient data mediation system (ADMS) from a traditional one :

- Data sources are heterogeneous, may be fixed or mobile and arbitrary connect and disconnect to the mediator during variable period of time
- The mediator can connect to the sources when and as long as there are visible over the wireless network
- The mediator should provide, for each application, the capabilities to define its data requirements in terms of events, context and user preferences.

Towards an Ambient Data Mediation System (2)

- The mediator should be able to aggregate data flows originated from different sources on the basis of rules specified by the application. Adaptors should hide the heterogeneity of data sources.
- The mediator should adapt itself to the current context, satisfy the user's preferences in terms of data sources, data delivery, relevance to domains of interest, privacy,...
- The mediator should be aware of energy consumption and manage consequently the connections to the sources and the usage of resources.

An Ambient Application

- A service oriented-architecture
- Events definitions and their queries
- ECA rules

The services

To offer interoperability among loosely coupled equipments, we assume a normalized schema composed of set of services. Services are viewed as virtual relations.

- `monitorTemperature() → [temp:{val:real, id_sensor:real}]` returns the temperature data and the sensor id that captured it.
- `getLocation(id_sensor:real) → location:{val:string}` returns the symbolic location associated with the senor.
- `turnHeat(location:string)` allows to activate the heating in the location given by the input parameter.
- `alarm()` triggers an intrusion alarm.

The Events

An event can be primitive or composite. Composite (or complex) events are combinations of simple or complex events using event operators. An event occurrence e_i is characterized by a set of attributes.

- **Temperature** :{value :real, timestamp :real, location :string}, $\mathcal{S}_{Temperature}$ = {monitorTemperature, getLocation}
- **PersonDetected** :{name :string, timestamp :real, location :string}, $\mathcal{S}_{PersonDetected}$ = {monitorPresence, getPerson, getRoom}.

For each event, a continuous query based on services (i.e., virtual relation) or composition operators events is defined.

The ECA rules

The first rule of the application turns the heat if the *Temperature* event has been detected in the same location as the user and if its value is under a certain threshold.

on	Temperature t
if	t.value $<$ 15 and t.location = Context_location
then	turnHeat(t.location)

The second rule triggers an alarm if the *PersonDetected* event concerns a person which is not authorized to be at this location.

on	PersonDetected p
if	p.name not in Auth_People.name
then	alarm()

The ADMS approach

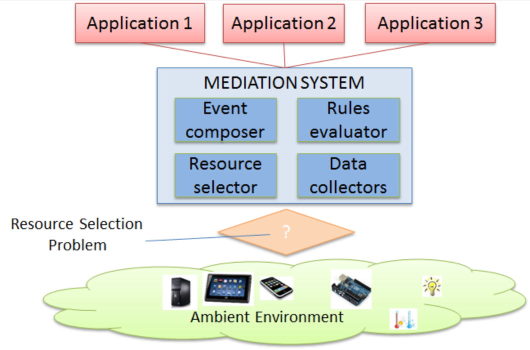


FIGURE : Ambient Mediation Framework

Preliminary definitions

- The application requirement
- The ambient environment

Definition of the application requirement

An *application requirement* $\mathcal{Q}$ represents the application interests regarding the ambient environment and is expressed as a set of required services $\mathcal{Q}$.

- $\mathcal{Q} = \{s_1, s_2, ..., s_n\}$ with $s_i \in \mathcal{S}$.

Based on the event specification and the ECA rules definition, the application requirement $\mathcal{Q}$ is derived. It contains the set of all the services used to instantiate the events, to evaluate the rule conditions or to execute the rule actions. For the example given above, the application requirement $\mathcal{Q}$ is :

- $\mathcal{Q} = \{monitorTemperature, getLocation, turnHeat, monitor Presence, getPerson, getRoom, alarm\}$.

Definitions of the Ambient Environment

An *active resource* is a resource which is visible, at a given instant and during a time interval, over the wireless network, and ready to provide a given service. Due to its energy consumption or to its distance to the user's location, an active resource may arbitrarily disconnect or be considered as useless.

- An *ambient state* A_s, $A_s = (R_a, l, t)$, is defined by the set of active resources with respect to a given user's location l and to a given timestamp t.
- An *ambient environment* Env is defined as a stream of ambient states, that is a potentially infinite sequence of states (R_a, l, t).

Problem Statement

The resource selection problem consists in successively and continuously computing Q over the ambient environment Env.

- computation of a best solution for Q
- decide at which frequency the computation should be done
- perform some adaptation between two execution cycles
- degrade the solution under a constrained execution

Evaluation of Q

At each instant t, the expected solution for Q is the set of relevant services as well as the corresponding resources over an ambient state.

1. $RR_a \subseteq R_a$ is the set of relevant resources, defined as : $RR_a = \{r \in R_a, \exists s' \in Q | match(r.s, s') = 1\}$
2. $\mathcal{P}(RR_a)$ is the powerset of RR_a, it is the domain on which Q will be interpreted.
3. $I_Q = \{i_Q \in \mathcal{P}(RR_a) | i_Q$ is a solution of $Q\}$. $I_Q \subseteq \mathcal{P}(RR_a)$. i_Q is a solution of Q if $\forall r \in i_Q, \nexists r' \in i_Q | r' \neq r$ and $match(r'.s, r.s) = 1$.

Solutions of Q

The selection of the best solutions $i_Q \in I_Q$ depends on two parameters :

- The *coverage* of the requirement Q, that is the ratio between the number of required services provided by the selected solution i_Q and the total number of services required by Q.
- The *disconnection probability* of each $r \in i_Q$, lower is this probability, higher is the capability of r to deliver data.

Solutions of Q

The process evaluates the previous subsets RR_a, I_Q and I'_Q at a **certain frequency** Δt and chooses only one solution in I'_Q in a random way.

- The computation of the best solutions among the powerset $\mathcal{P}(RR_a)$ must be made in a reasonable time, i.e., lower than Δt.
- Δt represents the delay between two execution cycles of the selection process.
- if it cannot be possible, it is necessary to give a degraded solution.

Adaptation

Resources continue to connect and disconnect between two execution cycles. To consider this problem, it is necessary to introduce an adaptation process which :

- Detect the changes at each acquisition of a new ambient state,
- Propose a new solution if the current solution is impacted by these changes.

For example, the resource disconnection may decrease the coverage of the current solution i_Q.

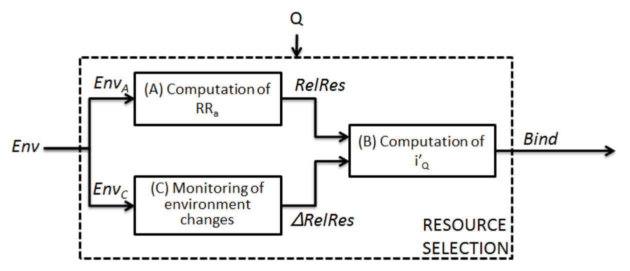
Adaptation

Two strategies can be used to solve this problem.

- Maintaining the coverage by using another solution $i'_Q \in I'_Q$ if it exists.
- Compute a new solution i'_Q from the current solution i_Q.

The goal is to maintain the coverage by substituting the disconnected resources by new ones and/or to increase it according to the new connected resources. This strategy allows to increase the coverage in some situations but is more costly in term of computation time.

The Resource Selection Problem Approach



The resource selection process uses a trigger-based mechanism.

- Each sub-process is triggered when an element is inserted into its input data structure (i.e., relation).
- a sub-process continuously verifies if there is an element into its input data structure and if it is the case, triggers its computation and consumes this element by deleting it from the relation.

Conclusion

- All algorithms have defined and implemented
- A prototype has been developped on ANDROID platform and ARDUINO sensor/actuator, communication through bluetooth.
- More Experiments have been conducted :
  - simulation methodology,
  - Δt has been measured,
  - different strategies have been compared

Questions ?