A Conceptual Framework for Development of Context-aware Location-based Services on Smart-M3 platform

Eldar Mamedov

P.G. Demidov Yaroslavl State University
Location-based services (LBS) are services oriented at mobile users that take current position of the user into account when performing their tasks.

**LBS and context**

LBS take into account only a particular part of the context, the location of the user.

If LBS will consider other aspects of a user context into account they could provide more relevant information and appropriate services for customers.

Examples of context-aware LBS:

- Routing for cars considering traffic conditions
- Routing for people considering the public transport location
The framework is designed to provide relevant objects from the database to the user taking his/her preferences and context into account.

### Framework foundations
- Smart-M3 platform
- Preference SQL system

### Smart-M3 platform
- **Semantic information broker (SIB)**
  The core component that stores shared semantic data.
- **Knowledge processors (KPs)**
  The active parts of a system which can provide, modify and query data.
Preference model

Preference definition

Preference is considered as a strict partial order on the domain values of attributes of the database relation.

Preference constructors

Base constructors:
- BETWEEN, POS, LESS_THAN, LOWEST, ...
- NEARBY, WITHIN, ONROUTE, ...

Complex constructors:
- Pareto preference (⊗)
- Prioritization preference (&)
Context model within framework

Context definition
Context is considered as a set of properties related to the user and representing his/her state.

Context types

- **Static user context**
  Contains user description that rarely changes. Usually stays the same throughout series of requests.

- **Dynamic user context**
  Contains user description that frequently changes. Could be different for each request and may change during the request processing.
Framework overview

Different kinds of input
- Request from the client service
- Corresponding user's context
- User profile

Main roles of the framework
- Client KP
- Context-aware preference term generator KP (CAPTGenerator KP)
- Preference query executor KP (PQE KP)
Data flow of the framework

Client KP

External services

Objects database

Context-aware preference term generator KP

Preference query executor KP

User context

User profile

<user_request>

<user_context>

<user_profile>

dynamic context

static context

objects
Data structures for request presentation and processing

- **UserRequest**
  - `objectType`: string
  - `processed`: boolean

- **UserContext**
  - Consists in
  - Contains dynamic context

- **UserPreferenceTerm**

- **ResultItem**
  - `position`: int

- **DBObject**
  - Consists in

- **UserProfile**
  - Consists in
  - Contains static user context

- **UserProfileItem**
  - Contains

- **User**
  - Has profile
  - Has static user context

- **DBObject**
  - Consists in

- **PreferenceTerm**
  - Consists in

Eldar Mamedov
A Conceptual Framework for Development of Context-aware LBS
Preference term representation

1. BetweenPreferenceTerm
   - lower : number
   - higher : number

2. AttributePreferenceTerm
   - attribute : string

3. POSPreferenceTerm
   - preferredSubset : set

4. PreferenceTerm
   - contains rightOperand, leftOperand

5. ParetoComposition
6. PrioritizedComposition
Client KP

- Adds an objects that describe the user: User, UserContext and UserProfile
- Forms a request by adding a UserRequest, UserContext and PreferenceTerm objects to the SIB
- Subscribes for the pattern (UserRequest ID, ’pqe:processed’, ’true’) to detect the moment when request is processed
- Displays results to the user when the request is processed

Eldar Mamedov
A Conceptual Framework for Development of Context-aware LBS
CAPTGenerator KP

- Adds a `ContextAwareGenerator` object
- Subscribes for the pattern (ANY, `’rdfs:class’, ‘pqe:UserRequest’) to track the addition of particular `UserRequest` objects
- When the `UserRequest` object is added and matches the KP specifications it reads dynamic context associated with the request and static context bound to the user and generates preference terms out of these data
- When the KP prepares the preference term from the context it adds a `ProcessedRequest` object containing this resulting term and associate it with the `UserRequest`
Subscribes for the pattern (ANY, 'rdfs:class', ContextAwareGenerator) to track the addition and removal of ContextAwareGenerator objects

Subscribes for the patterns (ANY, 'rdfs:class', UserRequest) and (ANY, 'pqe:generates', ANY) to track the addition of UserRequest and ProcessedRequest objects

Combines the user request, context and profile preference terms together by the following prioritization order

\(<\text{user\_request}> \& <\text{user\_context}> \& <\text{user\_profile}>\)²

Executes the resulting preference term on the object database, adds resulting list of objects to the corresponding UserRequest object and assigns 'true' to the value of the processed property of the UserRequest object

Conclusion

- Presented a conceptual framework for development of context-aware location-based services
- Designed an architecture of the framework based on the Smart-M3 platform
- Described main components and data structures of the framework