Chameleon: Context-Awareness inside DBMSs
Hicham G. Elmongui and Walid G. Aref (Purdue University)
Mohamed F. Mokbel (University of Minnesota)

Introduction
Context is any information that can be used to characterize the situation of an entity. Examples of context include, but are not limited to location, time, location, identity, and activity of a user. This paper proposes a general context-aware DBMS named Chameleon that will address the need for today's specialized database engines, e.g., spatial DBMS, temporal DBMS and heterogeneous DBMS given time, and identity can be treated as context in the general context-aware DBMS framework. In Chameleon, we will be able to combine multiple contexts into more complex ones why the proposed context composition, e.g., a heterogeneous DBMS that also provides spatial-temporal and location contextual filtering. By a proof of concept, we constructed two case studies using the same context-aware DBMS platform within Chameleon. One case study treats identity as a context to realize a privacy-aware (Hippocratic) database server while the other case study treats space as context to realize a spatial database server using the same proposed contracts and interfaces of Chameleon.

Classification of Contexts

<table>
<thead>
<tr>
<th>Contextual Relation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Context</td>
<td>order</td>
</tr>
<tr>
<td>Subject Context</td>
<td>invoice</td>
</tr>
<tr>
<td>Preference</td>
<td>books</td>
</tr>
<tr>
<td>Billing</td>
<td>address</td>
</tr>
</tbody>
</table>

Context-Awareness SQL Constructs
CREATE CONTEXT context_name (context_type) AS *

Conceptual Evaluation

Examples of context-aware applications include:

- User Context: User's preference to buy books
- Subject Context: ISBN, title, author
- Preference: Rating, preference, price

The user can specify his context using the context-aware SQL constructs.

Interfacing Hippocratic Databases

Using Chameleon, we limit both disclosure and retention of patients data in a healthcare facility as what happens in Hippocratic Databases, whenever a patient is admitted to the facility, he/she has to sign a privacy policy. The privacy policy specifies what information is to be released to which recipient. However, the policy also specifies for which purpose the information is to be released. On an opt-in basis, the healthcare facility also allows patients to choose if they want any of their personal information to be released to other recipients. By the end of the admission period, the patient should have fulfilled the purpose for which the data has been collected. After this period, different recipients cannot access the data.

Interfacing Spatial Databases

In Chameleon, we also have to model Spatial Databases. With the help of the context-awareness, we ensure both range and nearest-neighbor queries. We can access skyline queries as well. Consider a real-time database containing information about houses. The houses table has the following schema (id, address, price, city). An application developer is interested in finding some spatial queries to the database, but has no privilege to add the location of the house to the table. An object context is created to add the location of houses.

Object Context

CREATE CONTEXT object_context (context_name) AS *

Range Query

CREATE CONTEXT object_context.range(region) AS *

Skyline Query

CREATE CONTEXT object_context.skyline(region) AS *

Object context composition

A composite context may be composed from basic ones. Such composition may involve composing more than one context whose contextual relation is an ordering relation. We provide these mechanisms to resolve the conflict among the different orders of object imposed in these contexts.

- Using the ORDER BY clause
- Using ranking algorithm
- Using skyline algorithm