A Survey of Context Modelling and Reasoning Techniques

Bettini, Brdiczka, Henricksen, Indulska, Nicklas, Ranganathan, Riboni

Pervasive and Mobile Computing 2008 (submitted), 2010 (published)

Presented by Adil Yalçın
Some context for the paper

Papers cited by year

New York, Ranganathan
Italy, Bettini, Riboni
Germany, Brdiczka
Australia, Henricksen, Indulska
Content of the talk

- The Need for Context Modelling
- Properties of Context
- Modelling & Reasoning
  - Spatial
  - Situation
  - Uncertainty
- Object-role based models
- Ontology based models
- Hybrid models
Context Modeling and Reasoning

Software engineering for pervasive context-aware applications

- Reduce application level complexity
  - Provide a "context" back-end
  - Context is inherently complex already
- Improve maintainability, evolvability and inter-operability

*Question*: Does a generic / universal / efficient / easy to model context system exist? *Is it possible?*
Understanding Context

Heterogeneity of context

● High volume/real time but raw:
  ○ Sensors, not "meaningful"

● Derived context:
  ○ Likely to be updated less frequently
  ○ Meaningful for user and application

● User settings: few updates, direct

● Static (e.q. geo -databases)
Understanding Context

Relationships and dependencies
● Context data is not independent

Reasoning
● Higher level context
  ○ Raw & disconnected => meaningful & organized
● Change is (generally) what's important
  ○ From users/applications point of view

Efficient provisioning
● Decide relevant context
Understanding Context

Timeliness
- Past
  - Context histories, Q: volume? / summarization
- Current / Future

Imperfections
- Varying quality
- Conflicting / Incorrect
- Aging
- Incomplete
Understanding Context

Usability / application development

- Describe your application context easily
  - Relations in real world -> modeling -> bits
- Expressive power
- Runtime
  - Efficient/fast, compute-aware, scalable, stable, etc

Mobile

- Computation, sensors, spatial layout of context
- Sharing context & adaptive services
Spatial Models

Representing coordinates/location

- **Geometric**
  - x-y & reference axis / GPS / WGS 84 / ...
  - Relations are implicit in the 2D domain

- **Symbolic**
  - Room Number / cell tower ID / etc
  - Relations need to be coded into a database
  - Symbols can further be mapped to geometric data
    ■ But, not necessarily
Spatial Models

- Most models are fact-based.

Building a spatial ontology [1]:
- Tier 0: Physical reality, point->single value
- Tier 1: Observations, limited accuracy
- Tier 2: Map observations to a state of a specific object
- Tier 3: Socially constructed reality (admin-legal-inst. rules, object naming)
- Tier 4: Cognitive / reasoning level
  - Higher level context / situations
Spatial Models

Spatial queries [2]:
- Position *(Where is X?)*
- Range *(Find events near me)*
- Nearest neighbor *(Nearest restaurant, printer, ...)*

Other things to consider:
- Imprecise location / multiple sensors
- Filtering relevant context/data by location
- Interoperability of non-trivial location information
- Spatial data/compute partitioning
Situation Modelling and Reasoning

Situation: High level context
   For a state representation

Temporal/momentary state within context
   May become invalid but not necessarily form new situation

From raw sensor data to application/user domain

Change of situation -> adaptation

Not necessarily exhaustive of the whole world
Situation Modelling and Reasoning

Defining situations:

● Manually defined by designer
● Automatically recognized & machine learned
  ○ Ex: 4 layered learning framework [3]
    ■ Based on audio & video streams
    ■ Generates situations and roles/states of users
      ● An entity plays a role
  ○ Can achieve high accuracy in restricted test environments with small number of states
  ○ "Training period" : Often needs human in the loop / time consuming / data may not be available
  ○ Tradeoff between generalization and specification
Situation Modelling and Reasoning

Situation relationships

- Relationships can help reduce search space
  - + Performance, + Stability
  - - Requires exhaustive situation relation modeling
  - - Makes context description more complex.

Computation: Formal logic systems

- Situations as a set of conditions on context
- Con's:
  - Incomplete/uncertain data
  - Limited reasoning performance
    - Can use assertions under closed-world assumptions
Uncertainty - Modelling

- Imprecise
- Varying quality
- Conflicting / Incorrect
- Aging
- Incomplete ...

Modeling by information quality, with 6 axis [4]
- Coverage
- Resolution (by distance/time/symbolic)
- Accuracy
- Repeatability
- Frequency
- Timeliness (expiration time / decaying confidence)
Uncertainty - Reasoning

Goals:
- Improve context information
- Inter new kinds of information
- + Resolve conflicts

Uncertainty carries from lower (sensed) to upper (not-sensed) levels
Uncertainty - Reasoning

**Probabilistic Logic**
Associate events with probability, use proposition logic with complete axiomatisation.

**Fuzzy Logic**
Based on membership than probability. Operations include intersection / union / complement / modification. *Conceptual states for human-like reasoning.*

**Bayesian Networks**

**Hidden Markov Models**
Based on observations, find internal non-observed states. *Higher-level context.*

**Dempster Shafer Theory**
Theory of evidence based on belief functions and plausible reasoning. Often used for sensor fusion, from multiple observations of same phenomenon.

**Hybrid**: Use the best method given your context uncertainty properties
A Brief History in Context Modeling

- (Key,value) models
- Markup based models
  - Some of them XML based
- RDF based models
  - CC/PP (Composite Capabilities/Preferences Profile)

Limitations of the models above:
- A little too simplistic
- Limited reasoning options

So, improved models
- Ontologies / Object-role based models / Hybrids
Object-Role Based Models

- Fact-based model
- Support context queries & reasoning
- Provide support for software analysis & design
- Primarily based on Context Modeling Language (CML) (2002, [5])
Object-Role Based Models

CML (Context Modeling Language)
- Graphical notation (Similar to UML)
  - Direct mapping from model to implementation/runtime
- Supports different sources of context data
- Imperfections using quality metadata and "alternatives"
- Dependencies between context facts
- Context history (with constraints)
Sample CML Diagram

Key:
- ▲ Sensed fact type
- s Static fact type
- o Profiled fact type
- * Derived fact type
- [] Temporal fact type
- a Ambiguous/alternative fact type
- ← Key/uniqueness constraint
- → Snapshot uniqueness constraint
- → Alternative uniqueness constraint
- ↓ Dependency
- −− Quality annotation

* located near(p,d) iff located at(p, l1) and located at (d, l2) and l1 = l2
engaged in(p1,a) dependsOn located at(p2,l) iff p1 = p2
CML...

Querying
● SQL-like queries and simple assertions
● + uncertain information using a three-valued logic

Support for situations
● Based on a form of predicate logic
● + Efficient evaluation
● - Expressive power
  ○ Expressions are equality / inequalities / assertions
  ○ Situations / expressions can be combined
Ontology Based Models

Expressiveness vs complexity

Symbolic knowledge representation + optimized automated reasoning tools

+ Supports more complex context descr.
+ Share context among sources
+ Consistency checking
+ Derivation of situations

Models mostly based on OWL/OWL-DL
Ontology Based Models

\[ \text{BusinessMeeting} \sqsubseteq \text{Activity} \sqcap \geq 2 \text{hasActor} \sqcap \forall \text{hasActor.Employee} \sqcap \exists \text{hasLocation.(ConfRoom} \sqcap \text{CompanyBuilding)} \]

Focuses on building a decidable system.

- Limits on the complexity of domains
- Ex: No role-value-maps

Performance Problems

Extensions exist, but

- They increase already high \textit{computation cost} and some resulting languages are undecidable.
Hybrid Models

Aim for more flexible & generic systems.

Without sacrificing usability / worse complexity

Use multiple models
   Not by extending an existing model

No models can be made 'equal'
   Perfect translations do not exist between models

Loosely coupled reasoning procedures
## Hybrid Models

<table>
<thead>
<tr>
<th></th>
<th>Expressive</th>
<th>Efficiency</th>
<th>Inter-op</th>
<th>Problematic domains</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CML</strong></td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>Hierarchies / dominant contexts are harder to describe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Flat context types (atomic level)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Non-advanced uncertainty modeling</td>
</tr>
<tr>
<td><strong>Ontology</strong></td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓ ✓</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>Uncertainty</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Limitations in expressiveness and efficiency in current models</td>
</tr>
</tbody>
</table>
Proposed Hybrid Model

Fig. 3. Multilayer framework
Proposed Hybrid Model

Core ideas:

● Use simpler models to compute simpler context more efficiently
  ○ Even RDF or database based representations if possible

● Support ontological reasoning to derive complex information and for consistency checking

● Support inter-operability through a hidden layer (from application runtime)
References


References
