INTERACCIÓN 2011 Lisboa, Septiembre, 2-5

UCLM UNIVERSIDAD DE CASTILLA-LA MANCHA

Extensión a UsiXML para el Soporte del Awareness

José Figueroa Martínez[1]
Francisco Luis Gutiérrez Vela[1]
Víctor Manuel López Jaquero[2]
Pascual González[2]

Universidad de Granada[1]
Universidad de Castilla-La Mancha[2]

jfigueroa@ugr.es
fgutierr@ugr.es
victor@dsi.uclm.es
Pascual.Gonzalez@uclm.es

Outline

Introduction

Awareness, what it is?

UsiXML

UsiXML extension

- Awareness type definition
- Domain model extension
- Awareness Data Set and Requirement
- Awareness relations: Task and UI's

Conclusions and Future Work

Introduction

Problem, purpose and proposed solution.

The main problem: Non-existent Awareness support in UsiXML.

Our purpose*: D*efine the bases to integrate Awareness support in UsiXML.

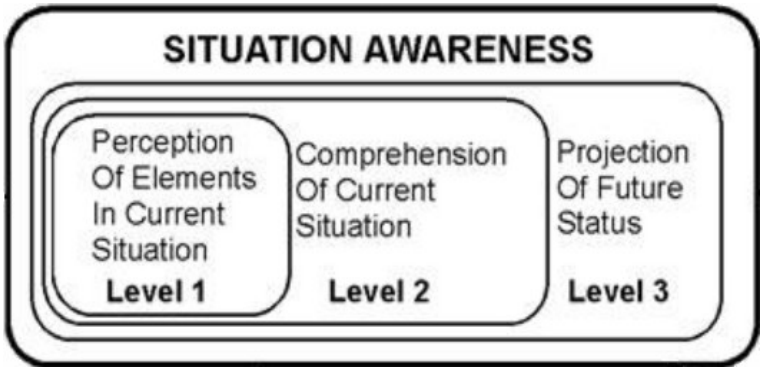
Our proposal*:* A set of Awareness models and extensions for UsiXML.

What is *missing*: The generation process required to complete Awareness support in UsiXML.

Awareness, what it is?

Theoretical basis.

"The *perception* of the **elements** in the environment within a volume of space and time, the *comprehension* of their meaning, and the *projection* of their status in the near future", Mica R. Endsley, 1995.



Conceptualization.

Elements: like attributes derived from other attributes.

Comprehension: Elements derived from other elements.

Projection: Operation over the element values.

UsiXML

User Interface Description Language and Model-Driven development methodology.

- Based on the Cameleon Framework.
- Include the steps for developing multi-modal and multi-context software systems.
- Many tools available, growing community, standarization on the way.
- Missing awareness support and collaborative ui's

UsiXML

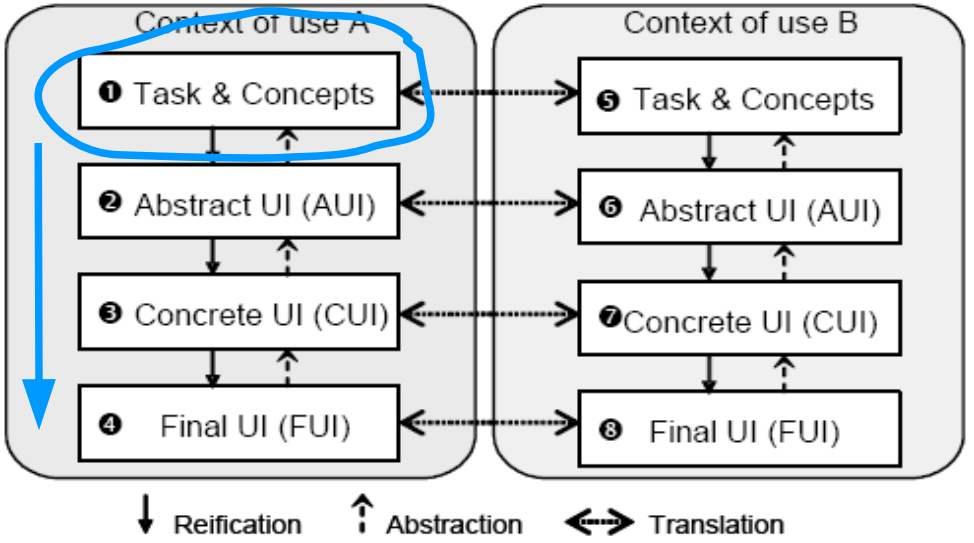
Development phases:

Task & Concepts
Tasks and domain concepts.

Abstract UI
Interaction spaces.

Concrete UI
AUI for specific context.

Final UI
Operational user interface.



Where awareness support belongs?

UsiXML extension

***First phase of Awareness Conceptualization ***

Additions

- Awareness models for defining and selecting awareness data.

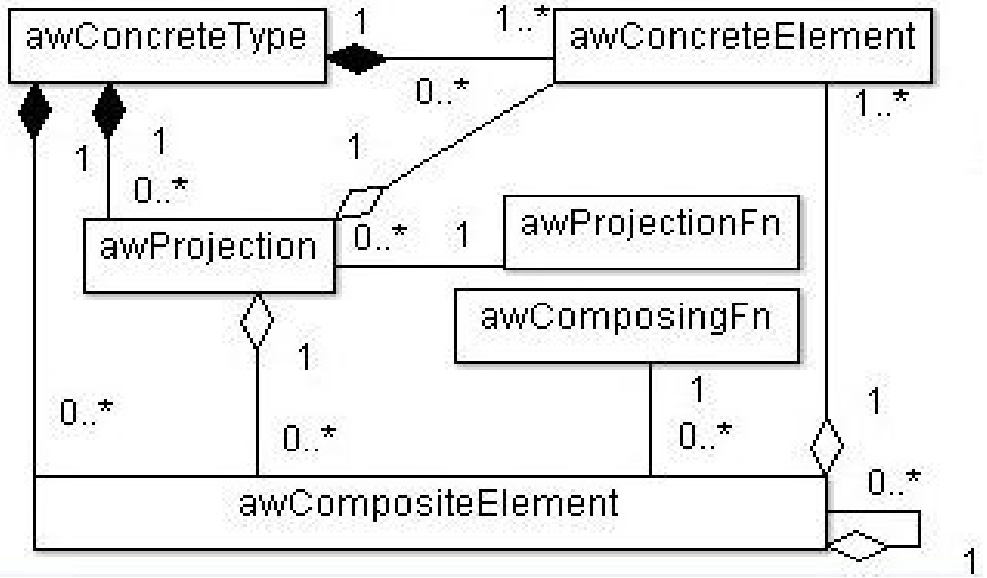
Changes

- Domain model: New entities for element's definition.
- Context model: New entities for runtime conditions.
- Mapping model: New mappings for traceability and linking.

Awareness type definition

We can define Awareness Types made of:

- **awConcreteElement**: represent awareness elements.
- **awCompositeElement**: represent the comprehension level.
- **awProjection**: represent the value's projection over time.



Example:

UserGeoLocationAwarenes : *awConcreteType*
- Identity *: awConcreteElement*
- Location : *awConcreteElement*
- Country: *awCompositeElement*
  - → Location (and external knowledge)
  - getCountryFromLocation : *awComposingFn*
- FutureLocation: *awProjection*
  - → Location
  - → Country
  - discreteProjectingFn : *awProjectionFn*

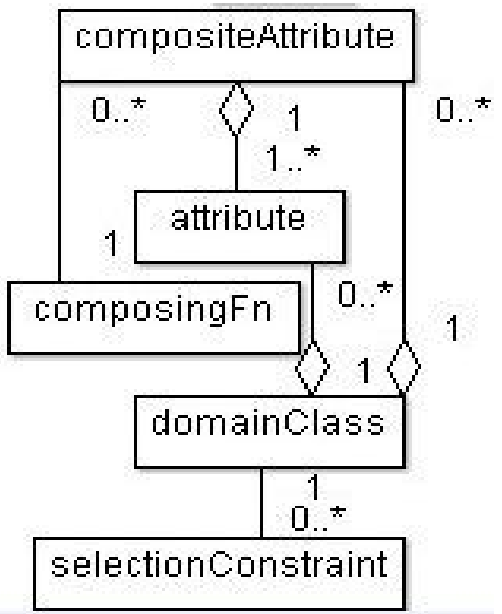
Domain model extension

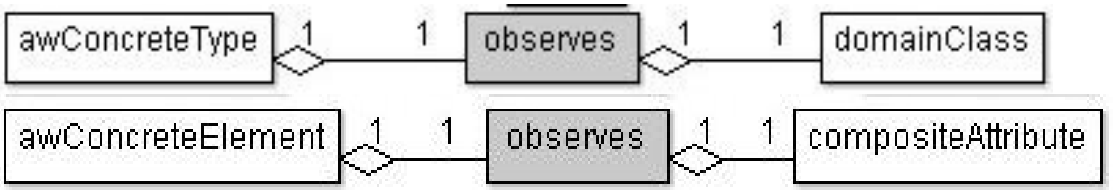
Awareness models are concretized from domain models through mappings.

compositeAttribute: allows composing attributes.

composingFn: self descripting.

selectionConstraint: allows creating instance selectors or data groups.





Concretizing Awareness Types**:**

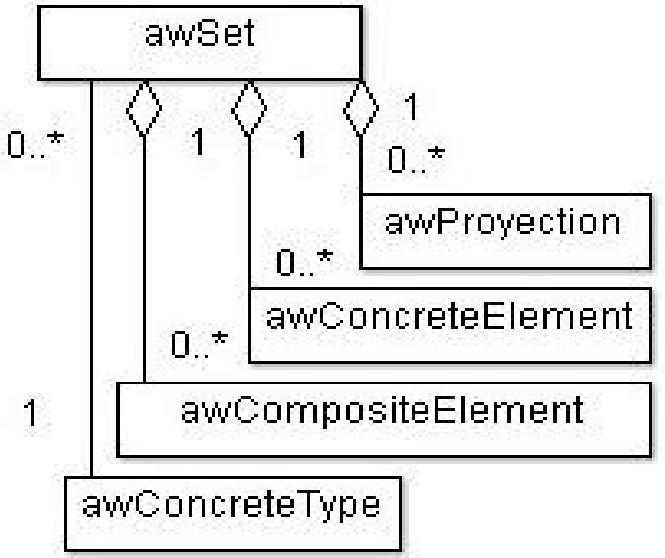
observes : *mapping*
→ UserGeoLocation : *awConcreteType*
→ User : *Class*

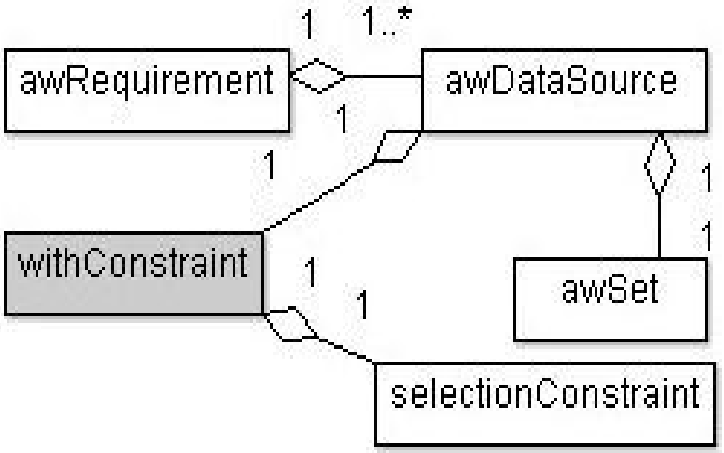
Concretizing Awareness elements:

observes : *mapping*
→ Identity : *awConcreteType*
→ Location : *awConcreteType*
→ UserLocation : *compositeAttribute*

Awareness Data Set and Requirement

- **awSet** is like a data structure linked to its data source.
- Used by the **awDataSource**, which defines a set of data with an specific structure.
- An **awRequirement** is just a group of **awDataSources**.



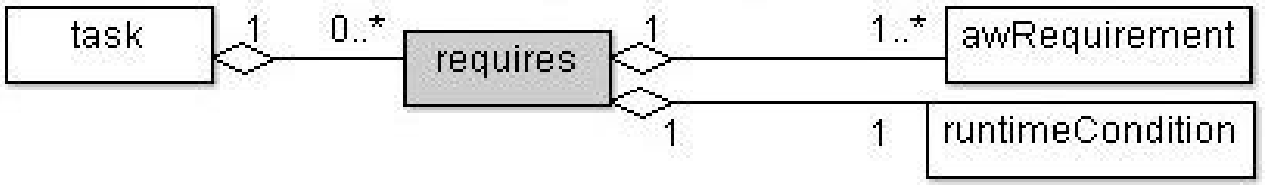


UserLocSet : *awSet*
→ Identity : *awConcreteElement*
→ Location : *awConcreteElement*
→ Country : *awCompositeElement*
→ UserGeoLocation : *awConcreteType*

UserFutureLocSet : *awSet*
→ Identity *: awConcreteElement*
→ FutureLocation : *awProjection*
→ UserGeoLocation : *awConcreteType*

UserLocReq : *awRequirement*
→ WorkingUsers : *awDataSource*
→ UserLocSet : *awSet*
→ withConstraint : *mapping*
→ OnlineUsers : *selectionConstraint*

Awareness relations: Task and UIs

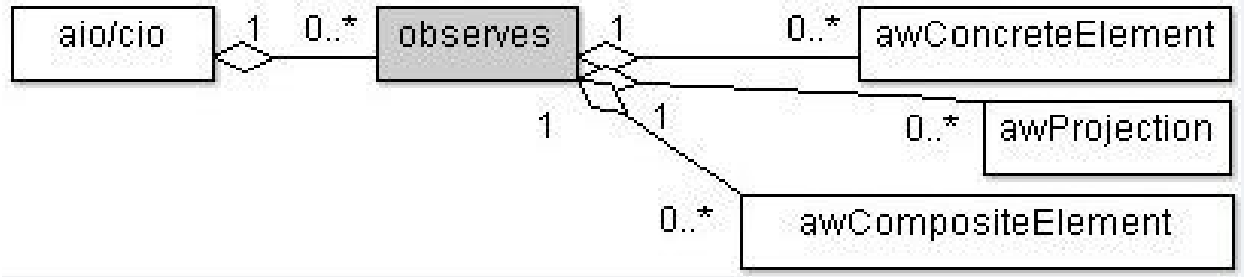


A Task can hold many Awareness Requirements with their runtimeConditions of applicability.

The runtimeCondition can be anything that could define an awareness requirement scenario.

Example:

coordinateGroup : *Task*
  requires
    → UserLocReq : *awRequirement*
    → ifUserAdmin : *runtimeCondition*



The observes mapping between awareness elements and aio/cios can be generated and used to validate mechanism.

Example:

coordinateGroupAio : *aio*
  observes
    → Identity
    → Location
    → Country

Conclusion and Future Work

We have defined an UsiXML extension for including Awareness as information requirement.

We need to integrate the transformation process to UsiXML's tools and Collaborative User Interfaces need to be added too.

Complete Awareness support is a must in collaborative systems and their developing tools. This is one step on it.

Questions ???