The Object Oriented Hypermedia Design Method

Dr. Daniel M. German

Department of Computer Science

University of Victoria

October 28, 2003 Version: 1.0.0
History

- Developed by Gustavo Rossi (Universidad Nacional de la Plata) and Daniel Schwabe (PUC Rio, Brazil)
- Its development started in 1995 and still continues
- Originally for Hypermedia, it has been adapted for the Web
They argue that:

- Good Web applications should be good hypermedia applications

- Traditional SEng methodologies do not contain useful abstractions to deal with the hypertext metaphor:
  - They do not provide the notion of linking
  - The interface of Web apps is more complex than in traditional software systems

- Navigation and functionality should be seamlessly integrated

- The navigational structure should be decoupled from the domain model of the app.
The Object-Oriented Hypermedia Design Method

The cornerstones of the approach are:

- Navigation objects are views (in the database sense) of conceptual objects
- The use of appropriate abstractions to organize the navigational space (navigational contexts)
- The separation of interface issues from navigation issues
Steps of OOHDM

Divides hypermedia design in 4 main activities:

- Conceptual design
- Navigational design
- Abstract interface design
- Implementation
Relationships between the different activities
# Summary of the Methodology

<table>
<thead>
<tr>
<th>Activities</th>
<th>Products</th>
<th>Formalisms</th>
<th>Mechanisms</th>
<th>Design Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual Modeling</td>
<td>Classes, subsystems, relationships, attribute perspectives</td>
<td>Object-Oriented Modeling constructs (classes, relationships, use cases)</td>
<td>Classification, aggregation, generalization and specialization</td>
<td>Model the semantics of the application domain</td>
</tr>
<tr>
<td>Navigational Design</td>
<td>Nodes, links, access structures, navigational contexts, navigational transformations</td>
<td>Object-Oriented Views Object-Oriented State charts Context Classes User centered Scenarios Design Patterns</td>
<td>Classification Aggregation, generalization and specialization</td>
<td>User's profile and task. Emphasis on cognitive aspects,</td>
</tr>
<tr>
<td>Abstract Interface Design</td>
<td>Abstract interface objects, responses to external events, interface transformations</td>
<td>Abstract Data Views Configuration Diagrams ADV-Charts Design Patterns</td>
<td>Mapping between navigation and perceptible objects. Composition and generalization/specialization</td>
<td>Model perceivable objects, implementing chosen metaphors. Describe interface for navigational objects Define lay-out of interface objects</td>
</tr>
<tr>
<td>Implementation</td>
<td>Running application</td>
<td>Those supported by the target environment</td>
<td>Those provided by the target environment</td>
<td>Performance, completeness</td>
</tr>
</tbody>
</table>
Conceptual Design

- A model of the application domain is built using well-known OO.
- Main concern is to capture the domain semantics as "neutrally" as possible:
  - Represents objects, their relationships and collaborations
  - Little concern for the types of users and tasks
  - If the app requires dynamic updates, this model will evolve into the object model for the application
- Products: a class schema built out of sub-systems, classes and relationships
- OOHDM does not prescribe any method to produce this schema
Conceptual Model for an online Magazine

Story
- Type: string
- Title: string
- Sub-title: string
- Date: date
- Summary: string
- Text: string
- Distribution: {local, national}

Related to

Person
- Name: string
- Bio: string
- Address: string
- Telephone: string
- URL: string
- Email: string
- Photo: image

Is Author of

Essay
- Illustration: [Photo, Video]

Translation
- Link to original
- Comments

Interview
- Sound
- Illustration: [Photo, Video]

Q&A
- Question: string
- Answer: string
Navigational Model

- It is built as a *view* over the conceptual model.
- Allows the creation of different models, according to different user profiles.
- Each navigational model provides a “subjective” view of the conceptual model.
Navigational Model (cont)

- Takes into account:
  - Which objects will be navigated
  - Composition structures (how we are creating composites)
  - What is the underlying structure of the navigation: in which contexts will the user navigate?
  - Whether navigational objects might look different depending on the context in which they are navigated
  - Which connections and access structures exist among the navigable objects
Products

- **Navigational Class Schema**: defines the navigable objects of the app (created from the conceptual schema)

- **Navigational Context Schema**: defines how navigable objects are clustered together and navigated
Navigational Class Schema

- Defines three types of navigational classes:
  - Nodes
  - Links
  - Access structures: they represent possible ways to access the
    nodes (i.e., indexes, guided tours)
Nodes Definition

```
NODE name [FROM className: varName] [INHERITS FROM nodeClass]
attri:  type1  [SELECT name1] [FROM class1:varName1, classj: varNamej
   WHERE logical expression]
attr2:  type2  [SELECT name2]...
..., attrn:  typen [idem]
END
```

Where
- `name` is the name of the class of nodes we are creating.
- `className` is the name of a Conceptual Class (from which the node is being mapped).
- `nodeClass` is the name of the super-class
- `attri` are the names of attributes for that class, `typei` the attribute’s types.
- `namei` are the subjects for the query expression and `vari` are mute variables used to express logical conditions.
- `-logical expression` allows defining classes whose instances are a combination of objects defined in the conceptual schema when certain conditions on their attributes and/or relationships hold.
Nodes Definition (cont)

NODE Story [FROM Story:St] [INHERITS FROM Person]
author: String [SELECT Name] [FROM Person:Pr WHERE Pr Is Author of St]
author_bio: String [SELECT Bio] [FROM Person:Pr WHERE Pr Is Author of St]
.... (other attributes “preserved” from the conceptual class Story)
toAuthor: Anchor (Is Author of)
END
Example of a Navigational Schema

**Story**
- Type: string
- Title: string
- Sub-title: string
- Date: date
- Summary: string
- Author: string [SELECT Name] [FROM Person:Pr WHERE Pr is Author of St]
- Author_Bio: string [SELECT Bio] [FROM Person:Pr WHERE Pr is Author of St]
- Text: string
- Distribution: {local, national}

**Essay**
- Illustration: [Photo, Video]

**Translation**
- Link to original
- Comments

**Interview**
- Interviewee: string [SELECT Name] [FROM Person:Pr WHERE Pr Grants St]
- Recording: Sound
- Illustration: [Photo, Video]

**Q&A**
- Question: string
- Answer: string
Navigational Context

- It is a set of nodes, links, context classes and other (nested) navigational contexts
- Defined by enumeration or by stating a property
- 5 different types:
  - Simple class based: each element of the context should satisfy a property (all sports stories)
  - Class based group: it is a set of contexts, of which each is a simple class context (list all contexts, one for each type of story)
  - Link based: the selection is based on a relationship, usually 1-to-n (all stories by Joe Smith)
Navigational Context (cont)

- Types of Navigational Contexts (cont)
  - Link based group: collection of link based contexts (all stories by author)
  - Enumerated: elements are explicitly enumerated (guided tour)

- Another category:
  - Dynamic: if they are defined by the navigation or interaction of the user (shopping basket, history, or user-modifiable database)
  - Otherwise static
Navigational Context Schema
Newer Developments

- Integration of processes
- UML