TOWARDS VARIABILITY MODELLING FOR REUSE IN HYPERMEDIA ENGINEERING

Peter Dolog and Mária Bieliková

{dolog, bielikova}@dcs.elf.stuba.sk

Department of Computer Science and Engineering
Faculty of Electrical Engineering and
Information Technology
Slovak University of Technology in Bratislava

Contents

- Motivation
- Related Work
- Requirements
- Background
- Our Approach
- Examples
- Conclusions and further work

Motivation

- Similar information can be reused in several courses
- Infrastructure components can also be reused for several courses
- To support
  - Reuse of information and software components in hypermedia application
  - To support generation and adaptation

Current approaches

- OOHDM
- UWE
- Petri nets (χTrellis)
- Hypercharts (XHMBS)
- AHAM

➢**To provide mechanism for:**

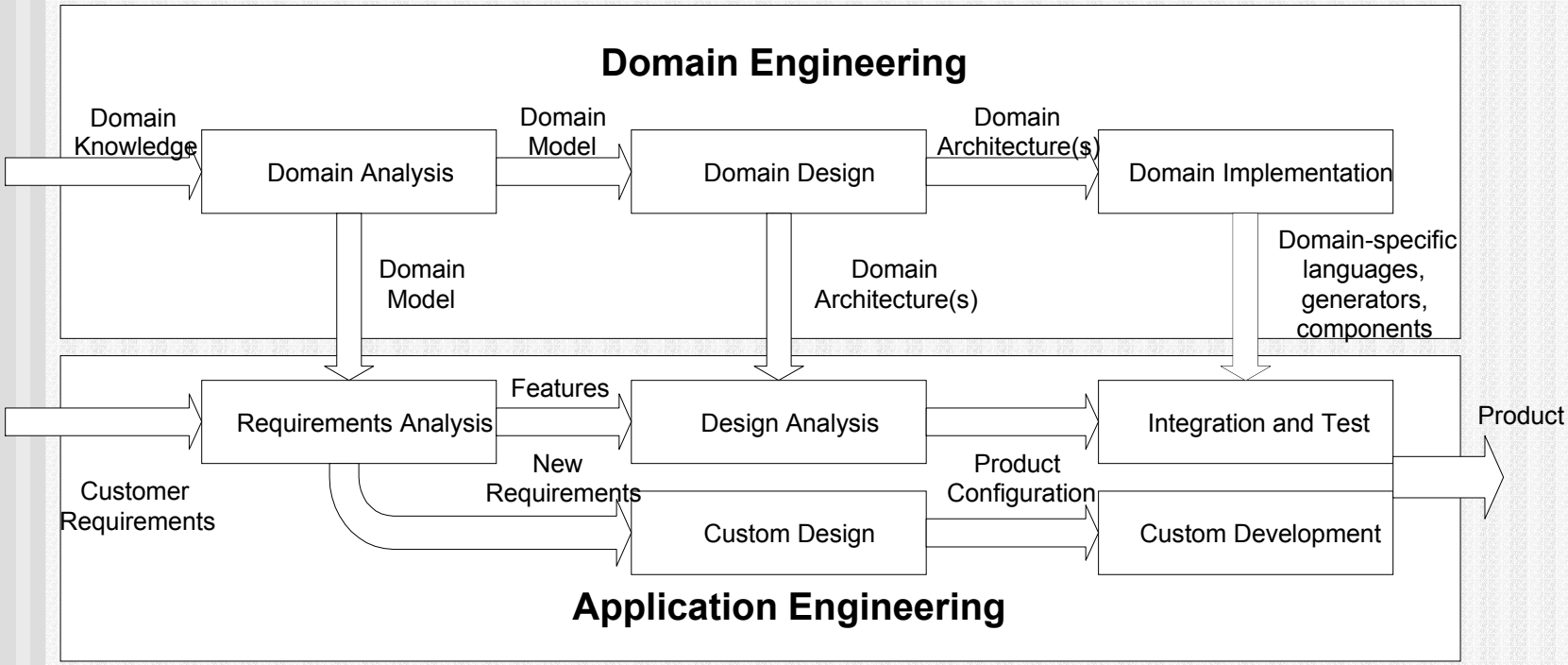
➢**Reuse in application family**

➢**Getting more than one navigation path and content organisation according to defined rules**

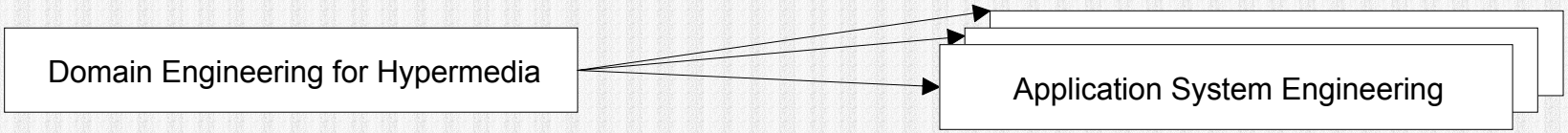
Requirements

- To handle variability of information and software features for hypermedia already at domain analysis level
- To map features and concepts to navigation path objects
- To automate some steps by introducing reusable components and models

Background


Domain Engineering
Domain Knowledge
Domain Analysis
Domain Model
Domain Design
Domain Architecture(s)
Domain Implementation
Domain Model
Domain Architecture(s)
Domain-specific languages, generators, components
Requirements Analysis
Features
Design Analysis
Integration and Test
Product
Customer Requirements
New Requirements
Product Configuration
Custom Design
Custom Development
Application Engineering

Our Approach



Define Conceptual Domain Model
- Real World
- Representation

Define Feature Model
- Real World
- Representation

Define Domain Architecture
- Information
- Navigation
- Presentation
- User
- Environment
- Adaptation

Implement Domain
- Reusable Components
- Tools
- Reuse Infrastructure
- Domain Specific Languages

Analyse Requirements of Stakeholders

Perform Selection of Reusable Components

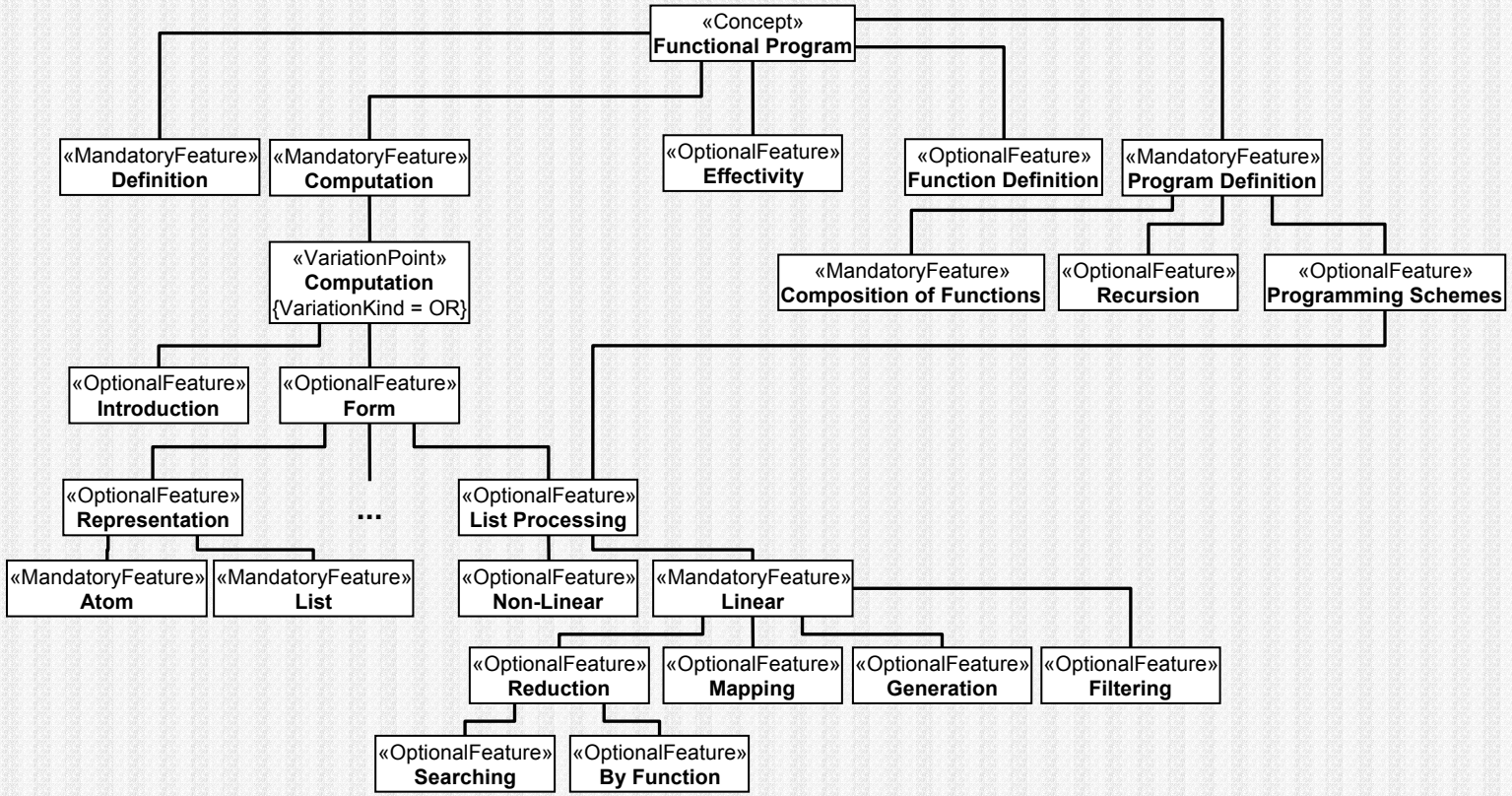
Define and Scope Requirements not Covered by Reusable Components

Do Design of Custom Components

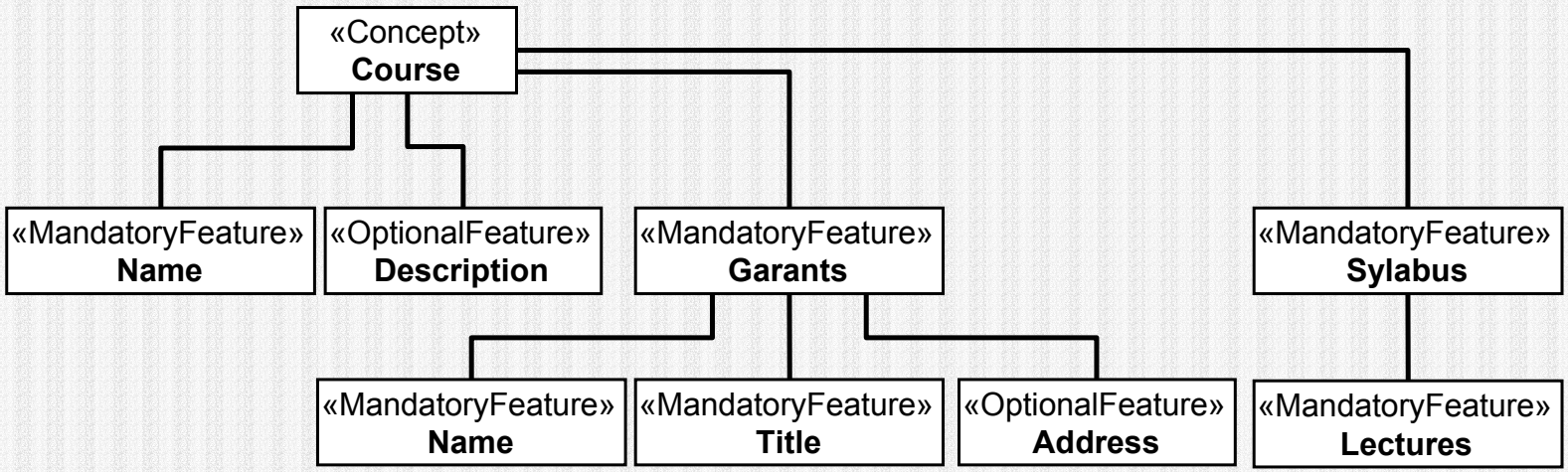
Specialise and Integrate Resusable and Custom Components

Consider Variability and Reuse at Reuse Time and at Bind Time

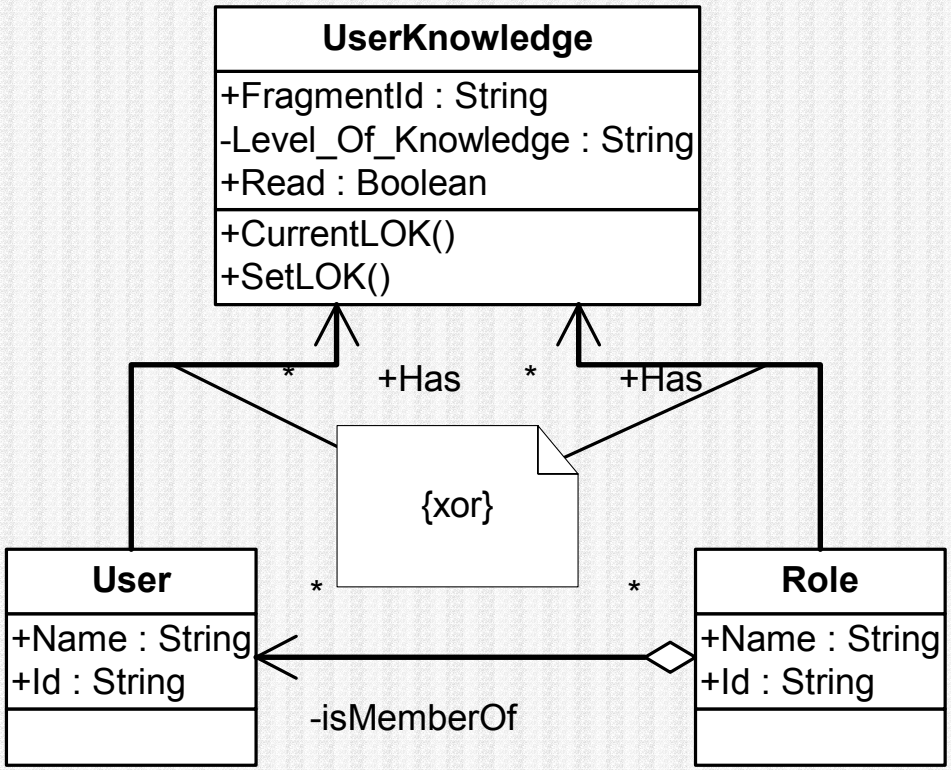
Domain Model – real world



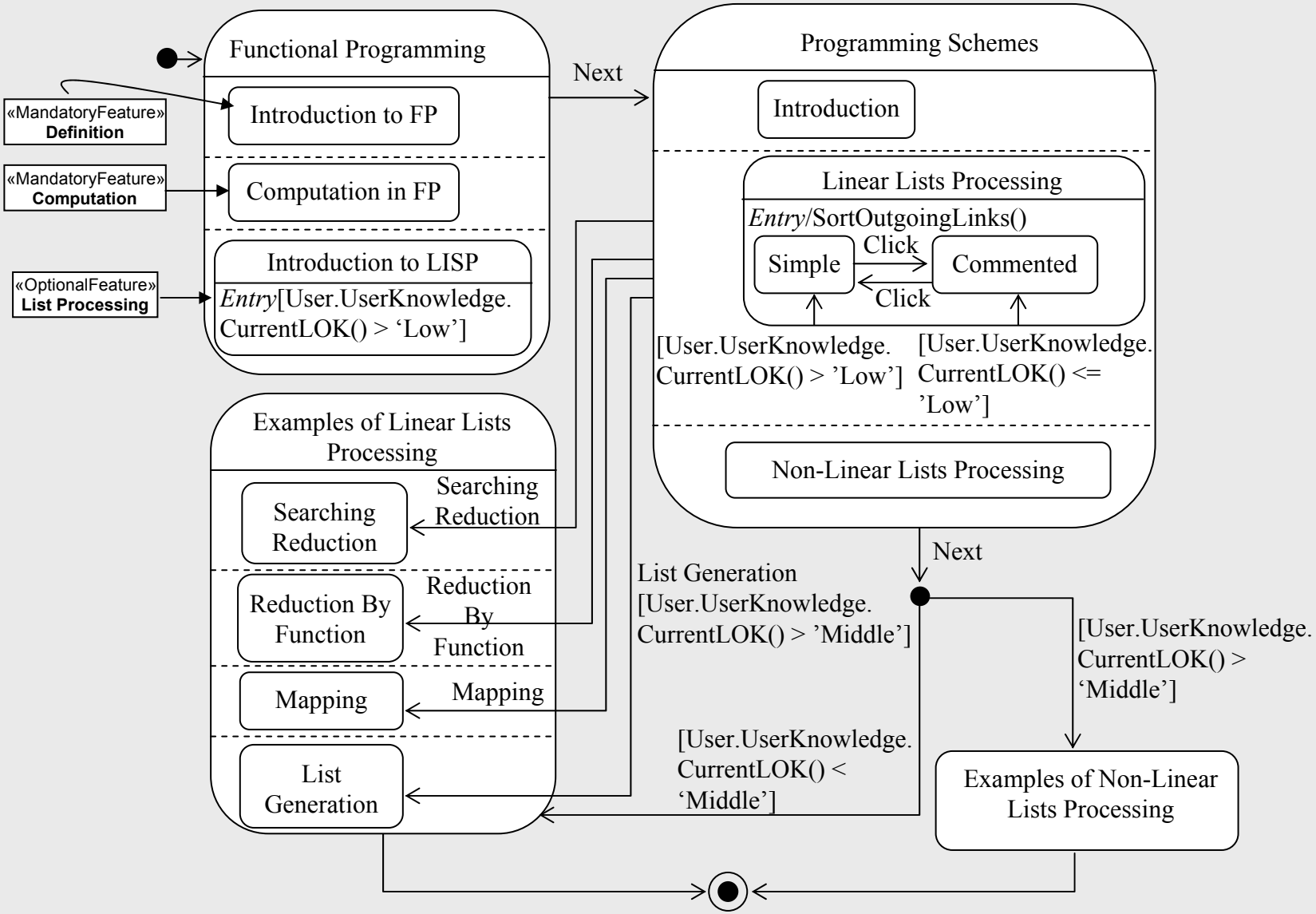
Domain Model - representation



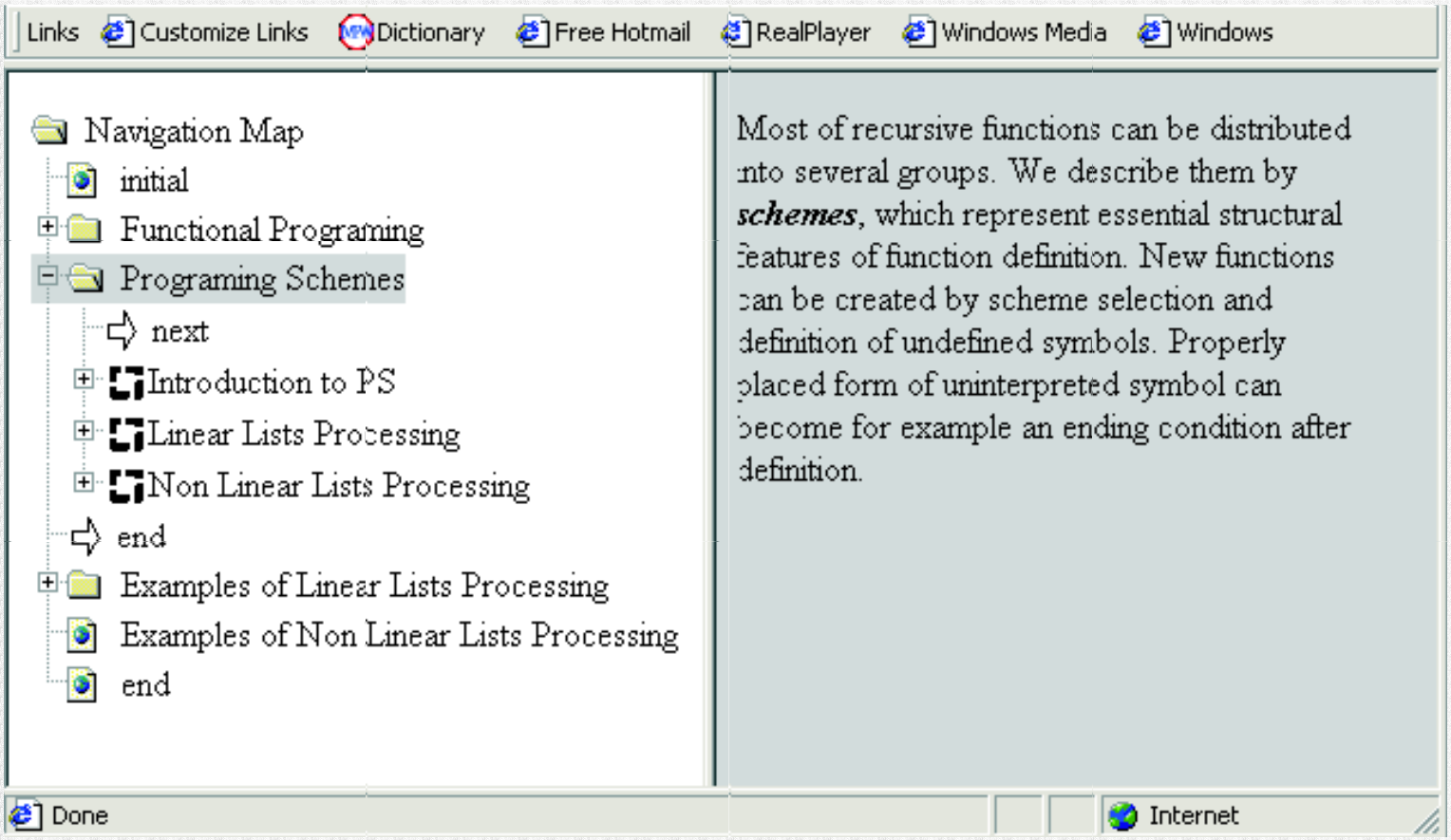
User Model


UserKnowledge
+FragmentId : String
-Level_Of_Knowledge : String
+Read : Boolean
+CurrentLOK()
+SetLOK()
*
+Has
*
+Has
{xor}
User
+Name : String
+Id : String
*
*
Role
+Name : String
+Id : String
-isMemberOf

Navigation Model



Prototype - generator



Prototype - XSLT

```
<xsl:template match="Behavioral_Elements.State_Machines.Pseudostate.kind">
        <xsl:variable name="myid" select=
        "../Behavioral_Elements.State_Machines.StateVertex.outgoing/
        Behavioral_Elements.State_Machines.Transition/@xmi.idref"/>

        <xsl:variable name="target_state" select=
        "//Behavioral_Elements.State_Machines.Transition
        [@xmi.id=$myid]/Behavioral_Elements.State_Machines.Transition.target/
        Behavioral_Elements.State_Machines.StateVertex/@xmi.idref"/>

        insDoc(foldersTree, gLnk(2, "<xsl:value-of select="@xmi.value"/>",
        "<xsl:value-of select="//Behavioral_Elements.State_Machines.CompositeState
        [@xmi.id=$target_state]/Foundation.Core.ModelElement.name |
        //Behavioral_Elements.State_Machines.State[@xmi.id=
        \$target_state]/Foundation.Core.ModelElement.name"/>"))

</xsl:template>
```

Conclusions and Further Work

- Domain engineering approach for hypermedia
- Feature modelling in domain analysis for hypermedia
- Proposal for transforming feature models to state diagram for navigation
- Implementation of state diagram transformation into navigation map

➢ **Methods for generation**

➢ **Evaluation**