A UML Profile for Dynamic Execution Persistence with Monitoring Purposes

Eladio Domínguez Murillo, Beatriz Pérez Valle, María Antonia Zapata Abad

University of Zaragoza (Spain)  University of La Rioja (Spain)  University of Zaragoza (Spain)

MiSE’13
May 19\textsuperscript{th}, San Francisco, CA
1. Introduction and background
2. Development of the DFMS Framework
3. The SEP Profile
4. Application and contributions of our approach
5. Conclusions and future work
1. Introduction and background
2. Development of the DFMS Framework
3. The SEP Profile
4. Application and contributions of our approach
5. Conclusions and future work
1. Introduction and background

Context

• In order to provide increasingly better services, best practice guides and standards for information technology guide organizations to follow a continuous process improvement.

• Organizations must be compliant with legislation.

• System monitoring facilitates the trace of processes, and helps to achieve processes improvement.
1. Introduction and background

Problems

Complex Data mining processes

System monitoring

Logs

event
timestamp
performer

...
1. Introduction and background

Our approach

• Defining **more complete persistence structures** that facilitate the subsequent processing of the system trace.

• **Our particular context:**
  - Easing audit and process improvement
  - Ensuring accuracy and consistency of data
  - Guiding the transformation process of the storage structures
1. Introduction and background
2. Development of the DFMS Framework
3. The SEP Profile
4. Application and Contributions of our Approach
5. Conclusions and future work
Decision Facts Management System (DFMS)

- Verifies clinical guidelines against quality properties.
- Develops guideline-based decision support systems (GBDSSs).
To guide the physicians during the application of a guideline in a very specific way in order to help in their decision making.

- **Traceability**: To automatically record the history of the application of the guideline to patients.

Can be used:
- as **key evidence** in a medical negligence case.
- as a **reference** in future encounters with the patient.
- as a **resource** for ongoing changes in the definition of guidelines over time.

2. Development of the DFMS Framework

**GBDSS for a guideline**
2. Development of the DFMS Framework

GBDSS for a guideline

• **Execution module**: corresponds to the Java implementation of the statechart.

• **Persistence component**: constitutes the main module which guarantees the persistence of the guideline application.
  
  • **Persistence configuration component**: defined to achieve the persistent task.
  
  • **Trace database**: physically stores the data generated during the guideline application and whose instances come from the running of the execution module.

• **Platform library**: provides standard services related to the implementation of the presentation and the data layers, serving as “glue” between the execution module and the persistence component.
2. Development of the DFMS Framework

Decision Facts Management System

2
Representation of CG by means of UML Statecharts within the DFMS Framework.

Verification Process

Decision Facts Management System Framework

Verification Process

Clinical Guideline (Statechart in electronic format)

Clinical Guideline (Natural Language)

Requirements (Natural Language)

Has associated

2
MDA Techniques

1
MDA Techniques

Legend

Manual Definition
Automatic Process
Modules Communication
Source
Items Manually Obtained
GBDSS's Module Automatically Generated
GBDSS's Module Manually Obtained

Execution Module
Persistence Component
Platform Library

GBDSS for the guideline
2. Development of the DFMS Framework

Decision Facts Management System

Legend
- Manual Definition
- Automatic Process
- Modules Communication
- Source

Items Manually Obtained
GBDSS’s Module Automatically Generated
GBDSS’s Module Manually Obtained

Clinical Guideline (Statechart)

Verification Process

2.2 MDA Techniques (Model to Model)

PIM Stereotyped Class Diagram

SEF Profile

2.1 MDA Techniques (Model to Text)

Execution Module

GBDSS for the guideline

Clinical Guideline (Natural Language)
2. Development of the DFMS Framework

Decision Facts Management System
Since UML 2.x does not provide a definition strategy for profiles, but simply presents the *UML Extensibility Mechanism* package with its components…

we have based on a mixture of:

1. The proposal given by Conallen [Cona00], which defines UML profiles starting with a brief description and then specifying its stereotypes, tagged values and constraints, and

2. the proposal given by Gogolla and Henderson [GoHe02], which describes stereotypes using a complete template definition.

Additionally:

- We have extended [GoHe02] so that our profile conforms to the *UML 2.4 metamodel*.

- the correct definition and use of our extension mechanism is assured by the *definition of the stereotypes’ constraints* in both *natural language* and *OCL*. 
3. SEP Profile

Definition. Stereotypes

28 stereotypes

Profile name

Profile name

Stereotype name

Properties

 SEP Profile

<<metaclass>>
Class

<<stereotype>>
Region
top[0..1]:Boolean

<<metaclass>>
Association Class

<<stereotype>>
TransitionNotBetweenStates
transitionKind:TransitionKind
event[0..*]:String
guard[0..1]:String
action[0..1]:String

(incomplete, overlapping)

<<stereotype>>
TransitionFromPS

(complete, disjoint)

<<stereotype>>
TransitionToFinalState

<<stereotype>>
TransitionFromInitialPS

<<stereotype>>
TransitionFromChoicePS

<<stereotype>>
TransitionFromShallowHistoryPS

<<stereotype>>
TransitionFromEntryPointPS

<<stereotype>>
TransitionFromJoinPS

<<stereotype>>
TransitionFromJunctionPS

<<stereotype>>
TransitionFromDeepHistoryPS

<<stereotype>>
TransitionFromEntryPointPS

<<stereotype>>
TransitionFromForkPS

<<stereotype>>
TransitionToInitialPS

<<stereotype>>
TransitionToChoicePS

<<stereotype>>
TransitionToShallowHistoryPS

<<stereotype>>
TransitionToEntryPointPS

<<stereotype>>
TransitionToJoinPS

<<stereotype>>
TransitionToForkPS

State

<<stereotype>>
SimpleState

<<stereotype>>
CompositeState

(incomplete, disjoint)

<<stereotype>>
NonOrthogonalCompositeState

<<stereotype>>
OrthogonalCompositeState

(complete, disjoint)
### 3. SEP Profile

#### Definition. Stereotypes

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base class</td>
<td></td>
</tr>
<tr>
<td>InfrastructureLibrary::Core::Constructs::Class</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

This stereotype is defined as an abstract stereotype and it is the base for all the concrete stereotypes representing states in the statechart. It has two direct derived stereotypes: SimpleState and CompositeState.

**Attributes**

The State stereotype has three properties:

1. **entry** [0..1]: String, which refers to the optional behavior that is executed whenever the state, from which the stereotyped class comes, is entered
2. **exit** [0..1]: String, which specifies the optional behavior that is executed whenever the state is exited,
3. **do** [0..1]: String, that refers to the optional behavior that is executed while being in the state.

**Constraints**

```plaintext
1  Context InfrastructureLibrary::Core::Constructs::Class
2  inv: self.isStereotyped("State") implies
3    let generalizationClass:
4      self.generalization.general and
5    let associatedClasses:
6      self.ownedAttribute.association.memberEnd.class asSet() excluding(self) in

7      generalizationClass forall(e| e.oclIsKindOf(Class) and e.isStereotyped("Region"))
8      generalizationClass size=1
9      associatedClasses forall(c: Class | (c.isStereokinded("TransitionNotBetweenStates")
10      or c.isStereotyped("TransitionBetweenStates")
11      or c.isStereotyped("Region") )
12      and not (c.isStereokinded("State"))
```
3. SEP Profile

**Definition. Stereotypes**

**Profile name**

SEP Profile

**Properties**

**Stereotype name**

State

**Region**

**Transition**

**Definition. Stereotypes**

This stereotype is defined as an abstract stereotype and it is the base for all the concrete stereotypes representing states in the statechart. It has two direct derived stereotypes: SimpleState and CompositeState.

**Attributes**

The State stereotype has three properties:

1. **entry [0..1]:** String, which refers to the optional behavior that is executed whenever the state, from which the stereotyped class comes, is entered.
2. **exit [0..1]:** String, which specifies the optional behavior that is executed whenever the state is exited.
3. **do [0..1]:** String, that refers to the optional behavior that is executed while being in the state.

**Constraints**

```oclnotation
1 Context InfrastructureLibrary::Core::Constructs::Class
2 inv: self. isStereotyped("State") implies
3 let generalizationClass:
4 self.generalization.general and
5 let associatedClasses:
6 self.ownedAttribute.association.memberEnd.class → asSet() → excluding(self) in
7 generalizationClass → forall(e| e.oclIsKindOf(Class) and e.isStereotyped("Region"))
8 generalizationClass → size()=1
9 associatedClasses → forall(c| Class | c.isStereoKinded("TransitionNotBetweenStates")
10 or c.isStereotyped( "TransitionBetweenStates")
11 or c.isStereotyped("Region")
12 and not(c.isStereoKinded("State"))
```
This operation returns a set containing the Stereotypes of the UML Element to which the operator is applied and all the Stereotypes inherited by such stereotypes.

**Context**: 

\[ \text{allStereotypes} : \text{set(Stereotype)} \]

**Post**: 

\[ \text{allStereotypes} = \text{self.extension.ownedEnd.type asSet()} \rightarrow \text{union (self.extension.ownedEnd.type.generalization.general.allStereotypes)} \]

This operation determines whether the UML Element to which the operator is applied, has a stereotype whose name is equal to the input name or, if it has a stereotype, one of whose ancestors’ name is equal to the input name.

**Context**: 

\[ \text{isStereokinded} : (\text{StereotypeName}: \text{String}) \rightarrow \text{Boolean} \]

**Post**: 

\[ \text{result} = \text{self.allStereotypes} \rightarrow \text{exists (s: Stereotype | s.name = StereotypeName)} \]

This operation determines whether the UML Element to which the operator is applied has a stereotype whose name is equal to the input name.

**Context**: 

\[ \text{isStereotyped} : (\text{StereotypeName}: \text{String}) \rightarrow \text{Boolean} \]

**Post**: 

\[ \text{result} = \text{self.extension} \rightarrow \text{exists (e: Extension | e.ownedEnd.type.name = StereotypeName)} \]

This operation returns the UML Stereotype Element applied to the context element and whose name is equal to the input name.

**Context**: 

\[ \text{obtStereotype} : (\text{StereotypeName}: \text{String}) \rightarrow \text{Stereotype} \]

**Post**: 

\[ \text{result} = \text{self.extension.ownedEnd.type asSet()} \rightarrow \text{select (s:Stereotype | s.name = StereotypeName)} \]
1. Introduction and background
2. Development of the DFMS Framework
3. The SEP Profile
4. Application and Contributions of our Approach
5. Conclusions and future work
The DFMS framework has been implemented as an Eclipse plug-in resulting in a first prototype.

- We have implemented our model-to-model and model-to-text transformations using two MDD-based Eclipse plug-ins (the ATL M2M transformation tool and the MOFScript M2T transformation tool), which have been integrated into the developed plug-in.
4. Application and Contributions of our Approach

Development of the DFMS Framework

We have implemented our approach using two MDA-based tools: ATL and MOFScript.
4. Application and Contributions of our Approach

Development of the DFMS Framework

- UML model to Java
- UML model to SQL
- UML model to Hibernate
4. Application and Contributions of our Approach

Advantages of the SEP Profile

- Stereotypes have been used as useful labels to help during the transformation process
4. Application and Contributions of our Approach

Advantages of the SEP Profile

- Stereotypes have been used as useful labels to help during the transformation process.

```java
1 split.Property::attributeToColumn(c:object){
2   //Declaration of variables...
3   //Create a column for primary keys
4   if(self.name.startsWith('ID')){
5     print('"-self.name+"')
6     self.primitiveTypeToSQLType()
7     println('not null')
8     println('constraint PK-'+c.reuseName()+' primary key ('-self.name+')')
9     aux.add('+self.name+')
10    indexConsUnique.put(c, aux)
11 }
12 //Create columns related to association properties in the class diagram which will be foreign keys
13 else if (self.association !=null){
14   self.association.memberEnd -> forEach pp:split.Property| pp.name != self.name{
15     theOtherRoleProperty=pp
16   }
17 //Particular case 1: Association properties which come from the
18 //composite properties in composite state
19 if(c.hasStereotype("NonOrthogonalCompositeState") or c.hasStereotype("OrthogonalCompositeState")
20   and self.name.startsWith("roleLastStateInRegion")){
21     println('"-foreignKey.name+"')
22     foreignKey.primitiveTypeToSQLType()
23     println('null')
24   //Gather information in collections
25   //Variable used in `createConstraintsInCreateTable` rule for defining Uniqueness constraints
26   //for foreign keys (FKs) in 'Create table'
27   if(forConstraintsInCreateTable.get(c)==null){
28     aux2.add(self)
29     forConstraintsInCreateTable.put(c, aux2)
30   }
31   else{
32     forConstraintsInCreateTable.get(c).add(self)
33   }
34 //Variable used in `FKeyConstraints` rule for creating FKs in 'Alter table' instructions
35 if(forForeignKeys.get(c)==null){
36
```
4. Application and Contributions of our Approach

Advantages of the SEP Profile

- Specific stereotypes are transformed into concrete SQL statements in the database (foreign keys and triggers).

```java
var nombreTabla; 
var nombreTabla;
var jerarquia;
for (forEachKey) 
if (c.name.equals) 
    nombreTabla
else 
    jerarquia;
for (forEachKey) 
if (c.has) 
    nombreTabla
   jerarquia;
} 
println('s
if (es.name) 
    println();
else 
    if (jerarquia.print
```

- Generating foreign keys using `alter table` statements.

```sql
-- Constraints Section

   alter table AbsPacienteAplicandoProtocoloBRC52R1State add constraint Abs
   foreign key (numHistoriaClinica)
   references Paciente
   on delete cascade deferrable initially deferred;

   alter table AbsPacienteAplicandoProtocoloBRC52R1State add constraint Abs
   foreign key (IDEpapbrcs2s)
   references PacienteAplicandoProtocoloBRC52State
   on delete cascade deferrable initially deferred;

   alter table AbsPacienteAplicandoProtocoloBRC52R1State add constraint Abs
   foreign key (IDEpapbrcs4s)
   references PacienteConInfeccionSeguraS4State
   on delete cascade deferrable initially deferred;

   alter table AbsPacienteAplicandoProtocoloBRC52R1State add constraint Abs
   foreign key (IDEpapbrcs3s)
   references PacienteConSospechaBRC53State
   on delete cascade deferrable initially deferred;

alter table AbsPacienteAplicandoProtocoloBRC52R1State add constraint Abs
foreign key (IDT5)
references t5;
```

Generated constraints
4. Application and Contributions of our Approach

Advantages of the SEP Profile

- Ensures accuracy and consistency of data
  - Specific semantics of the stereotypes have been implemented as concrete constraints and triggers in the trace database.

- Guides the transformation process
  - We have defined the M2M and the M2T transformations (from the PIM to the persistence component) in such a way that they use the stereotype names as useful marks to help during the transformation process.

- Eases audit and process improvement
  - By means of the semantics provided by the stereotypes of the SEP profile, we obtain a more complete trace of the system, represented in the persistence structures of the trace database.
  - Such a trace constitutes a meaningful source of the information generated during the application of the guideline to the patient, easing enhanced auditing processes.
It is used in a Spanish hospital and has been developed on the basis of a guideline published by the US Agency for Health Care Research and Quality (AHRQ) National Guideline Clearing House (NGC).

- It has been defined mainly for diagnosis and prevention.

- **Representation:** text document of 10 pages, written in natural language with tables and flowcharts.
4. Application and Contributions of our Approach

IRC Guideline

Statechart Representation

* SSCHS: Study of semiquantitative cultures of catheter hub and skin
4. Application and Contributions of our Approach

IRC Guideline

2.2 From the Statechart to the PIM Stereotyped Class Diagram

Transformation rules

SEP Profile

Orthogonal state

PatientWithoutCVCandWithoutTreatment

PatientWaitingForCVCRemoval

PatientWaitingForMakiResults

PatientWithMakiResults

t34

PatientWaitingForCVCRemoval

PatientWaitingForMakiResults

t35: actionCarriedOut("CVCRemoval") /orderClinicalTest("Maki")

PatientWaitingForMakiResults

PatientWithMakiResults

t36: testResultsArrive("Maki") /assignTestResults("Maki")
4. Application and Contributions of our Approach

2.2 From the Statechart to the PIM Stereotyped Class Diagram
4. Application and Contributions of our Approach

From the Statechart to the PIM Stereotyped Class Diagram

The trace database comprises 152 tables.
The persistence configuration module comprises 151 mapping files, together with the hibernate.cfg.xml configuration file.
4. Application and Contributions of our Approach

- The DFMS framework has been implemented as an Eclipse plug-in resulting in a first prototype.
  - We have implemented our model-to-model and model-to-text transformations using two MDD-based Eclipse plug-ins (the ATL M2M transformation tool and the MOFScript M2T transformation tool), which have been integrated into the developed plug-in.

- The DFMS framework has been satisfactorily applied to several real-life guidelines used in different contexts within the medical care system, obtaining different GBDSSs working prototypes.
  - The guidelines used as case studies are:
    - IRC Guideline,
    - OPC Guideline: a clinical guideline for the management of obesity in primary care,
    - AP Guideline: a laboratory guideline to carry out the aliquoting process.
1. Introduction and background
2. Development of the DFMS Framework
3. The SEP Profile
4. Application and Contributions of our Approach
5. Conclusions and future work
5. Conclusions and Future Work

Conclusions

- We have given an overview of the definition and development of our DFMS Framework.

- We have presented the definition of the SEP profile:
  - It contributes to ease audit and process improvement.
  - The implementation of the semantics of its stereotypes helps to ensure accuracy and consistency of the data stored in the database.
  - Stereotype names have been used as useful marks to help during the transformation process.
Future work

- To apply our approach to other frameworks in which statecharts are also used to represent the system behaviour.

- To apply our approach to other dynamic modeling languages that may be used to represent guidelines.

- To use our system trace proposal with a specific standard or best practice guide for information technology.
A UML Profile for Dynamic Execution Persistence with Monitoring Purposes

Questions?

MiSE’13
May 19th, San Francisco, CA