Departamento de Eletrônica e Sistemas

From i* to OO-Method: Problems and Solutions

Fernanda Alencar, Beatriz Marín, Giovanni Giachetti, Emanuel Santos,

Oscar Pastor, Jaelson Castro, Xavier Franch

Agenda

- The Problem
- The Proposal
- Relating i* and OO-Method Approaches
- Some Problems and Solutions
- Conclusions and Future Works

The Problem

- Goal-Oriented Requirements Engineering (GORE) stood out because it is mainly concerned with the stakeholders intentions and their rationales
  - How to go from requirements models to the corresponding software product is still an open question.
  - We need a requirements model with such a structure that facilitates the specification of model transformations for the automatic generation of conceptual models used in MDD approaches.

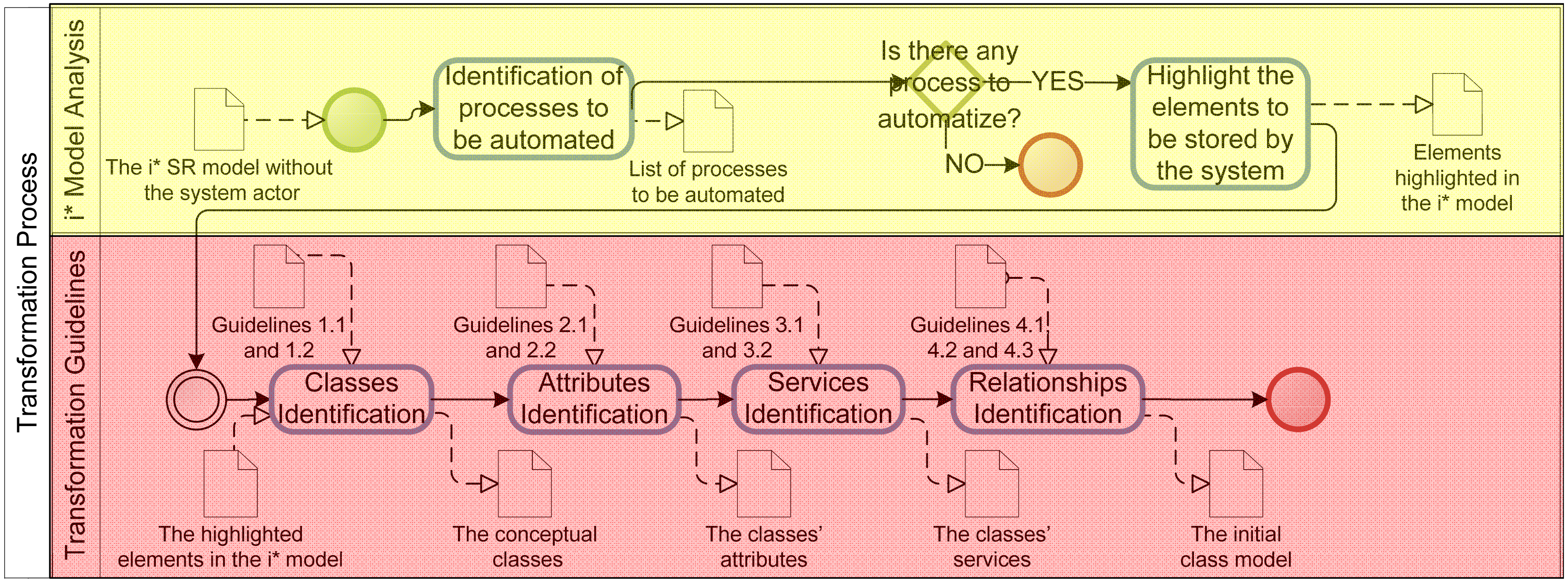
The Proposal

- We report on lessons learnt with a collaborative project[1], which aims at relating i* and the OO-Method approaches.
- Among the several GORE works, we have chosen the *i** framework because it is a consolidated modeling technique with good tool support, and an abstract syntax formalized by a metamodel specification.
- The *OO-Method* is used as a reference MDD technology because it has been successfully applied to industrial software development by means of the *OlivaNova* suite.

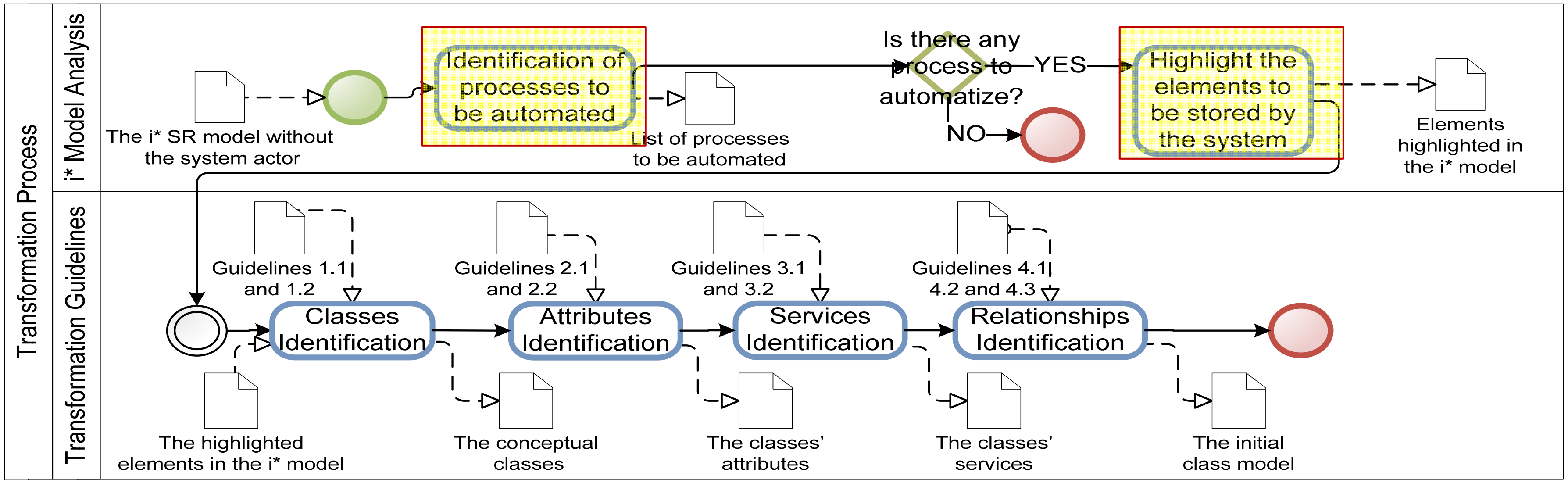
[1]CAPES-DGU: Integration of Organizational Modelling Techniques to Software Automatic Generation: OO-Method Case (in Portuguese). 2nd partial report. Ministério da Educação, Coordenação Geral de Cooperação Internacional Programa Brasil-Espanha da CAPES/DGU. Processo Nº 167/08, Brazil, 2010

Relating i* and OO-Method Approaches

The transformation process modeled with BPMN


Transformation Process
i* Model Analysis
The i* SR model without the system actor
Identification of processes to be automated
List of processes to be automated
Is there any process to automatize?
YES
NO
Highlight the elements to be stored by the system
Elements highlighted in the i* model
Transformation Guidelines
Guidelines 1.1 and 1.2
Guidelines 2.1 and 2.2
Guidelines 3.1 and 3.2
Guidelines 4.1 4.2 and 4.3
Classes Identification
Attributes Identification
Services Identification
Relationships Identification
The highlighted elements in the i* model
The conceptual classes
The classes' attributes
The classes' services
The initial class model

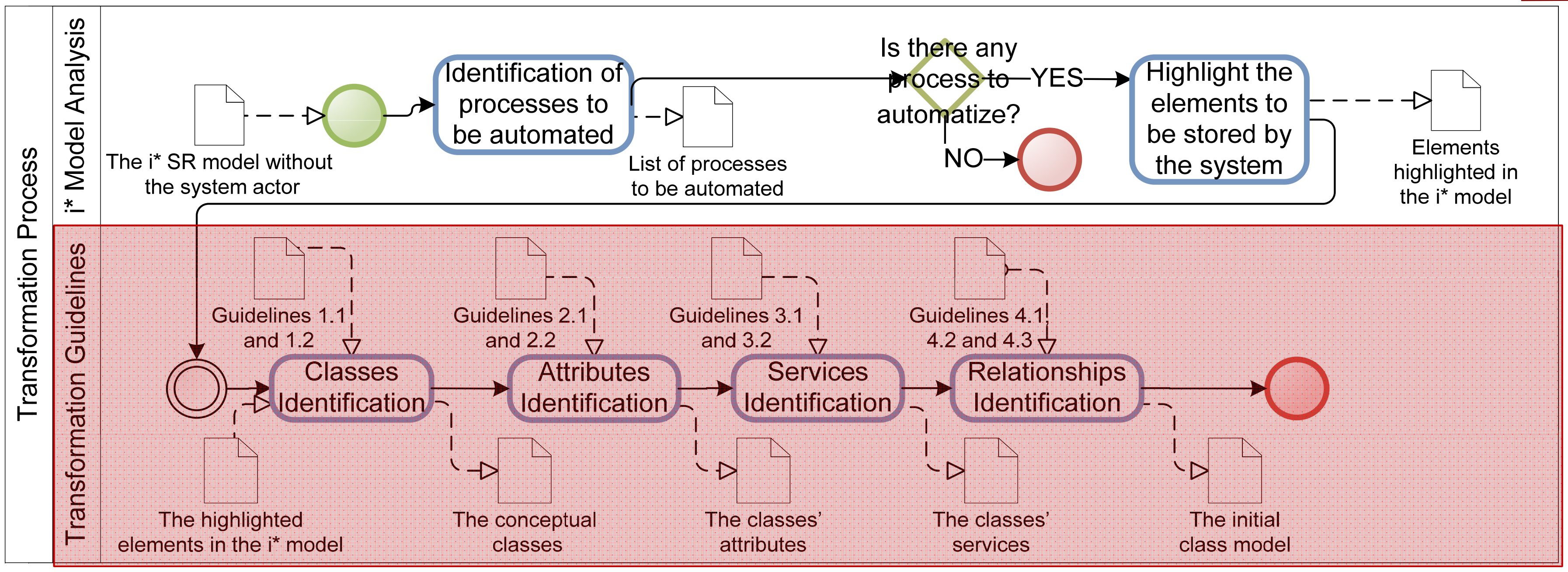
Relating i* and OO-Method Approaches



Transformation Process
i* Model Analysis
Transformation Guidelines
The i* SR model without the system actor
Identification of processes to be automated
List of processes to be automated
Is there any process to automatize?
YES
NO
Highlight the elements to be stored by the system
Elements highlighted in the i* model
Guidelines 1.1 and 1.2
Guidelines 2.1 and 2.2
Guidelines 3.1 and 3.2
Guidelines 4.1, 4.2 and 4.3
Classes Identification
Attributes Identification
Services Identification
Relationships Identification
The highlighted elements in the i* model
The conceptual classes
The classes' attributes
The classes' services
The initial class model

- Initially, we analyze the goals defined in the Early SR model in order to capture the organizational processes that we want to automate.
- Then, if there is any process to be automated, we highlight the intentional elements that are related to these processes (goals and tasks in the *i* model).*

Relating i* and OO-Method Approaches


Transformation Process
i* Model Analysis
The i* SR model without the system actor
Identification of processes to be automated
List of processes to be automated
Is there any process to automatize?
YES
NO
Highlight the elements to be stored by the system
Elements highlighted in the i* model
Transformation Guidelines
Guidelines 1.1 and 1.2
Guidelines 2.1 and 2.2
Guidelines 3.1 and 3.2
Guidelines 4.1 4.2 and 4.3
Classes Identification
Attributes Identification
Services Identification
Relationships Identification
The highlighted elements in the i* model
The conceptual classes
The classes' attributes
The classes' services
The initial class model

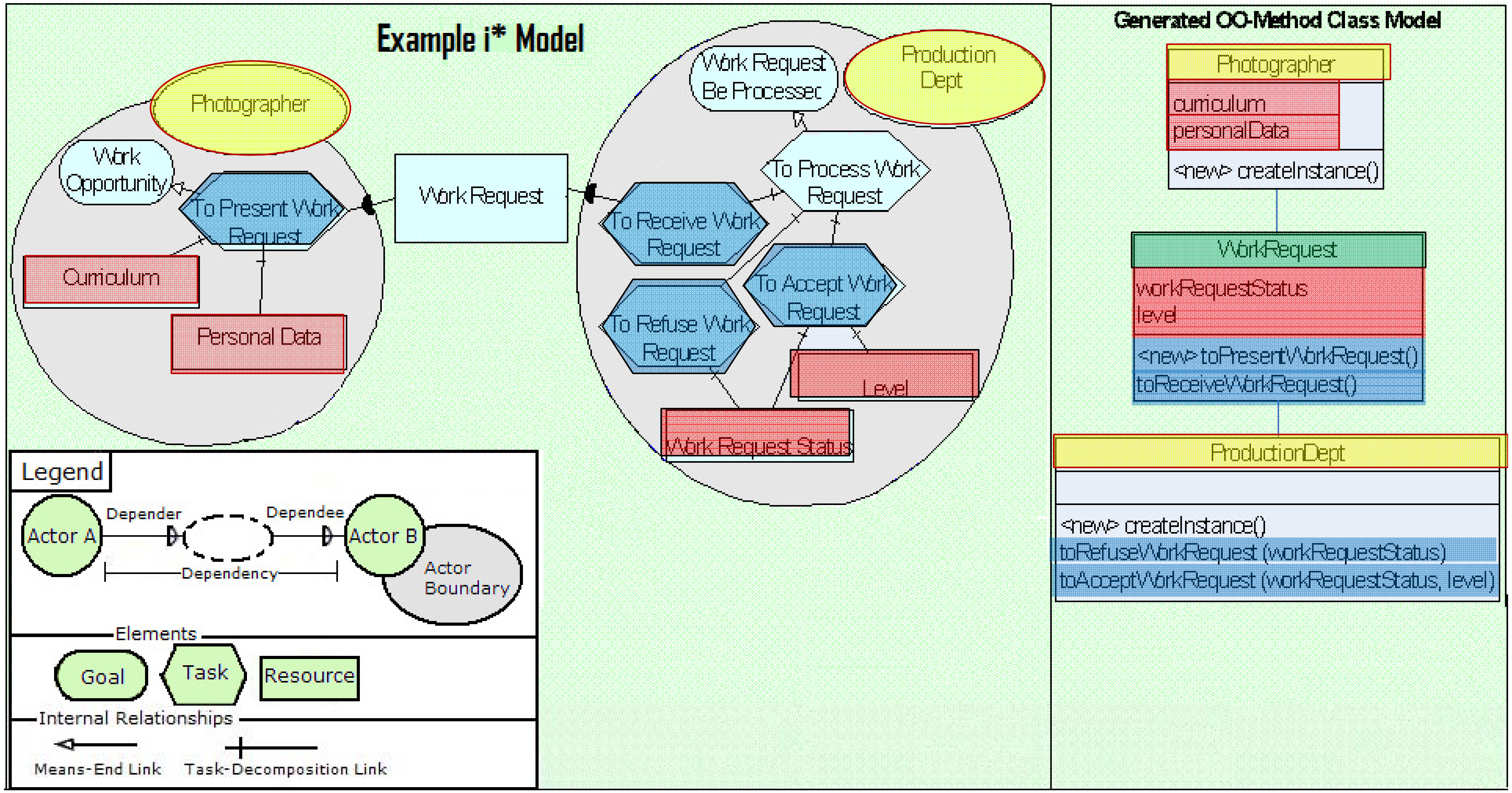
- From the list of identified intentional elements we obtain an initial skeleton of OO-Method conceptual model through the application of a set of transformation guidelines .

Relating i* and OO-Method Approaches

The Table depicts a summary of the transformation guidelines

i* **Construct**	**Additional Information**	**Class Model Construct**
Actor		Class
Resource	Physical entity	Class
	Informational entity related to a physical resource or an actor	An attribute that represents information of the class generated from the actor or physical resource
	Resource in a decomposition tree	Input arguments for the service generated from the related task
	Dependum resource	Input argument of the depender task
	Physical entity inside of an actor boundary	An association between the classes generated from the physical resource and the owner actor
Task	Participating in a resource dependency as depender or dependee	A service of the class generated from the dependum resource
	If generates a resource	A creation service of the class generated from the resource
Dependency link	Where the *dependum* resource and the *depender* and *dependee* actors are transformed in classes	Associations are automatically defined among the generated classes

Relating i* and OO-Method Approaches



Only those elements that are related to the intended system are considered.

Some Problems and Solutions

- **Problem 1.** *It is not possible to automatically infer if a resource corresponds to a physical or an informational entity*
  - **Solution.** *We propose to extend resources with an attribute which defines the its type because we pretend.*
- **Problem 2.** *Differences in the Abstraction levels of i* and OO-Method.*
  - **Solution.** *One possibility is to define an auxiliary model to record the traceability data.*
- **Problem 3.** *Two or more kind of elements of the i* model can be transformed into the same kind of element of the OO-Method class model.* In other words, the traceability between the conceptual representation of the system and the corresponding requirement element is lost.
  - **Solution.** *One possibility is to define an auxiliary model to record the traceability data.*

Some Problems and Solutions

- **Problem 4.** *Some relevant information of the i* model may be lost in the transformation process.*
  - **Solution.** *The intermediate model can also store the mapping required to identify these elements from the generated class model.*
- **Problem 5.** *It is not possible to directly specify which elements of the i* model must be automated.*
  - **Solution.** We propose to use a metamodel extension mechanism to label the corresponding *i** model, for instance, such a UML profile.
- **Problem 6.** *The cardinalities of the associations between classes cannot be automatically inferred.*
  - **Solution.** We propose the introduction of a new property in the i* model that allows the cardinality of the association among the generated classes to be automatically inferred. In fact in the context of Software Product Line development we have already proposed an i* extension that deals with cardinality (the so called i*-c)

Conclusions and Future Works

- The proposal defines guidelines which be automated as well as some procedures which are semi-automatic or even manual, i.e. require human intervention.
- The solutions presented are oriented towards the fully automation of the process.
- We want to minimize the dependency on highly experienced analysts and designers to manually transform the requirements models into appropriate OO-Method models.
- We are also working in proposal to evaluate the quality of requirements models

Conclusions and Future Works

- We plan to apply the transformation guidelines to different case studies in order to evaluate the correctness and completeness of our proposal.
- We plan to formalize and automate the guidelines using metamodeling standards (such as MOF [12]) and model-to-model transformations technologies (such as ATL [8]).
- We also consider the definition of metamodel extensions for the *i** framework in order to improve the modeling facilities for MDD environments and to completely automate the transformation of GORE models.