Integrating Information Systems Components: A Situation-Driven Approach

Nicolas Arni-Bloch, Jolita Ralyté,
Michel Léonard

UNIVERSITÉ DE GENÈVE

Context

Observation

- Complexity and diversity of IS application domains increase
- Cost of "from scratch" engineering is very high
- Reuse of existing solutions
  - Global approach
    - Total integration.
  - Application approach
    - Flexibility
    - Can better comply to the work process

Context

Problem

- Global approach
  - Lack of flexibility
  - Difficulty to extend
  - Work process has to comply to the system
- Application approach
  - Lack of integration
  - Need of trick, as glue, mediators, exchanges of messages, adaptors, translators, wrapper
  - Not well adapted to IS reality

⇨Our proposal

- Define the notion of IS-COTS and
- Specify the requirements for a new approach supporting IS-COTS integration.

Outline

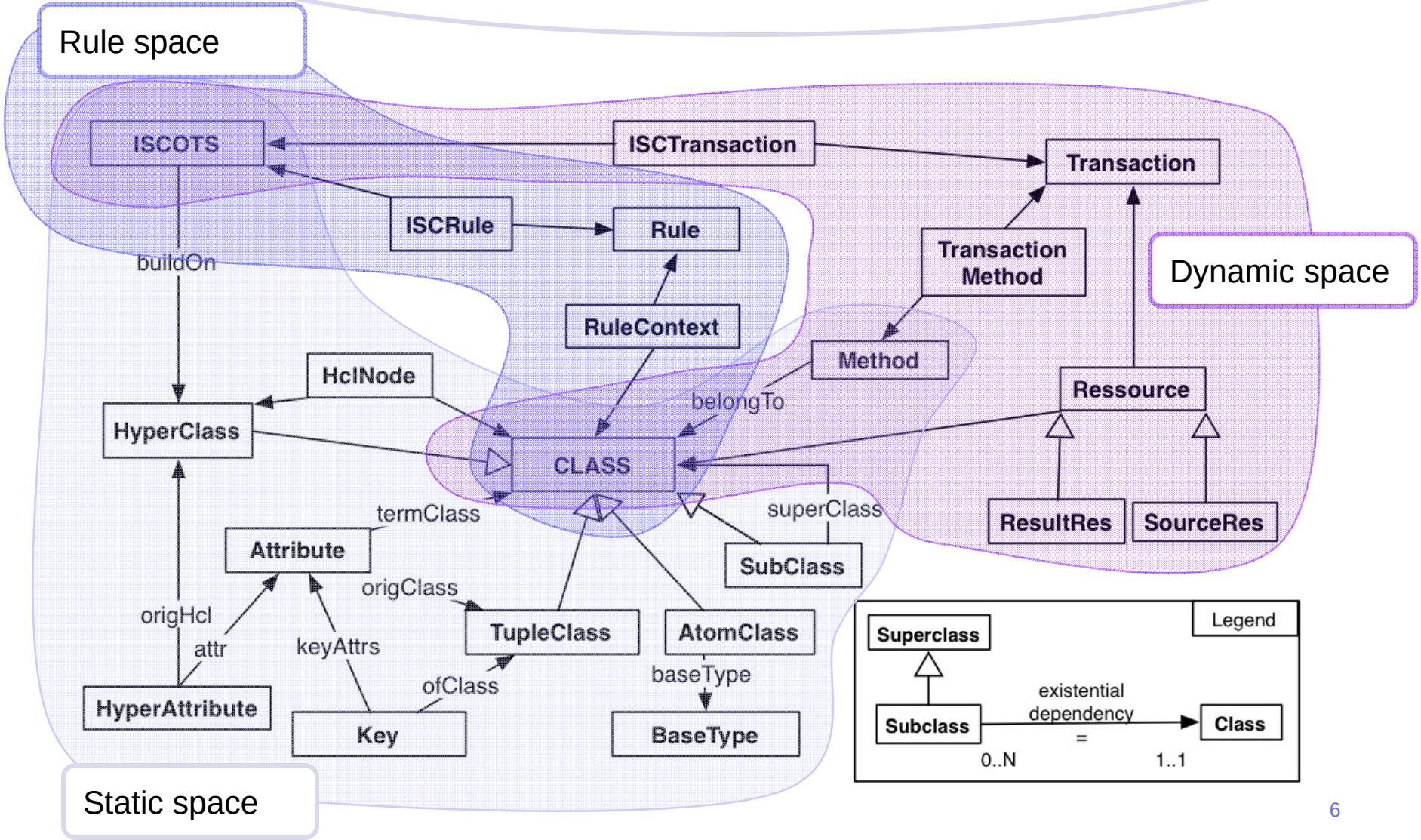
- IS-COTS Definition
- IS-COTS Example
- Requirements for a new approach supporting IS-COTS integration
- Method Chunk (MC) Template
- MC Examples
- MC Application Examples
- Conclusion

IS-COTS overview

<SS, DS, RS>

- SS static space:
  - Object-oriented model
  - Hyperclass concept
- DS dynamic space:
  - Bipartite net
    - One type of node is classes
    - Other type of node is transaction
  - Represent complete cycle of data processing
- RS rule space:
  - Preserve coherence, correctness and consistency.
  - Integrity rules, business rules.

IS-COTS metamodel

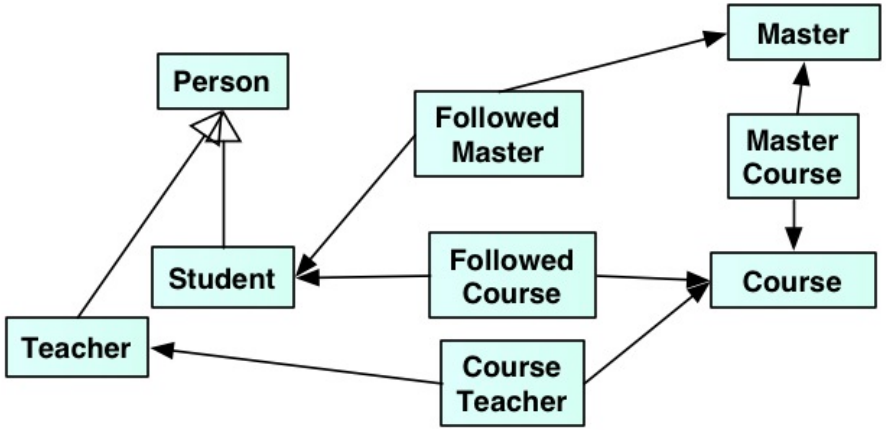


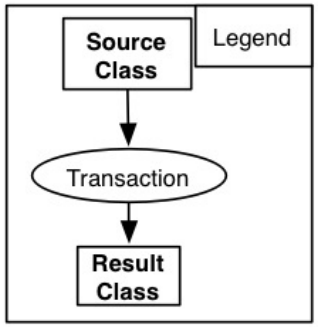
Example of IS-COTS

IS for Bachelor Diploma Management

IS-COTS for Master Diploma Management

QuickTime™ et un décompresseur TIFF (LZW) sont requis pour visionner cette image.





- Allocate students to courses
- Store examination results
- Associate teachers to courses
- Generate documents

Requirement$_1$

Method Requirements Specification

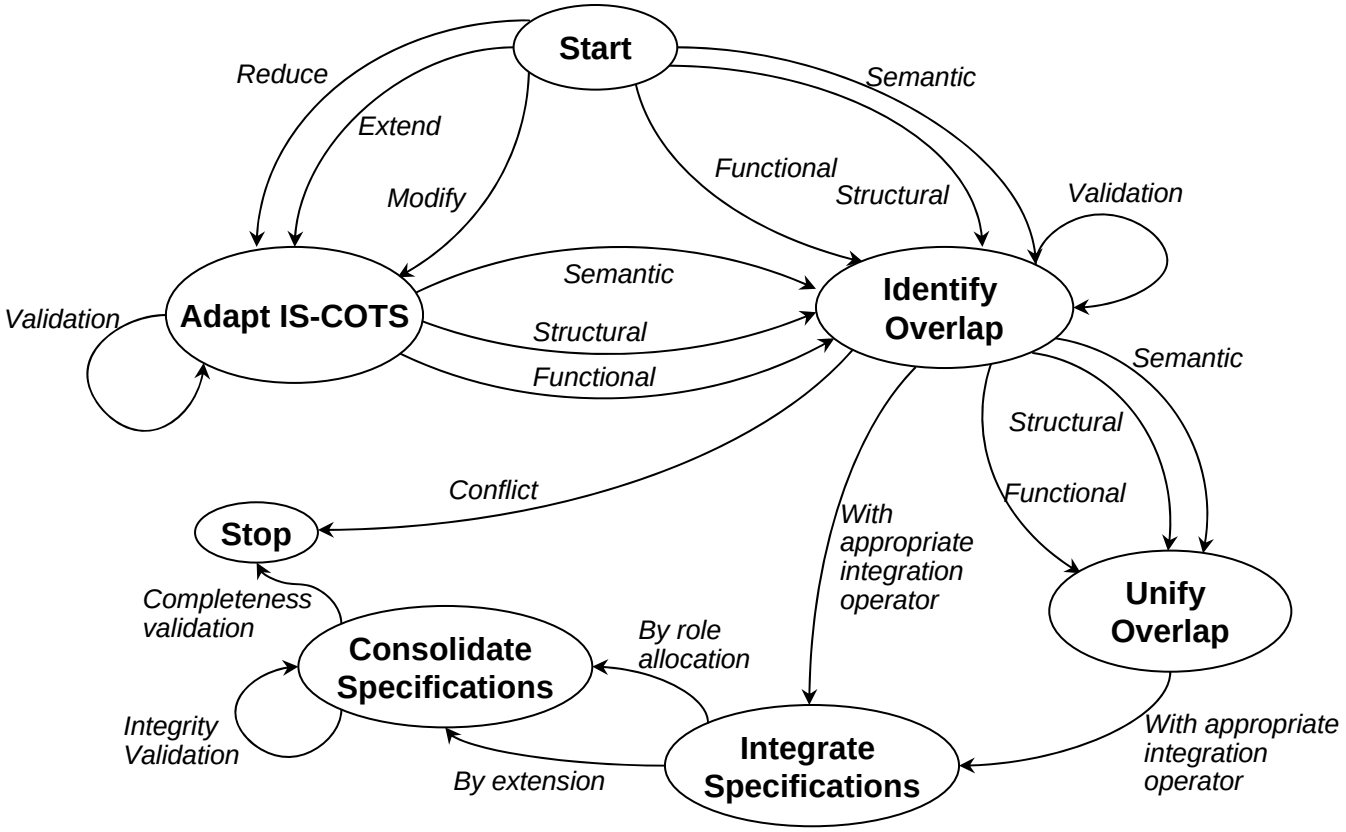
Objective

⇒ To define a situational method as a collection of reusable method chunks for IS-COTS integration.

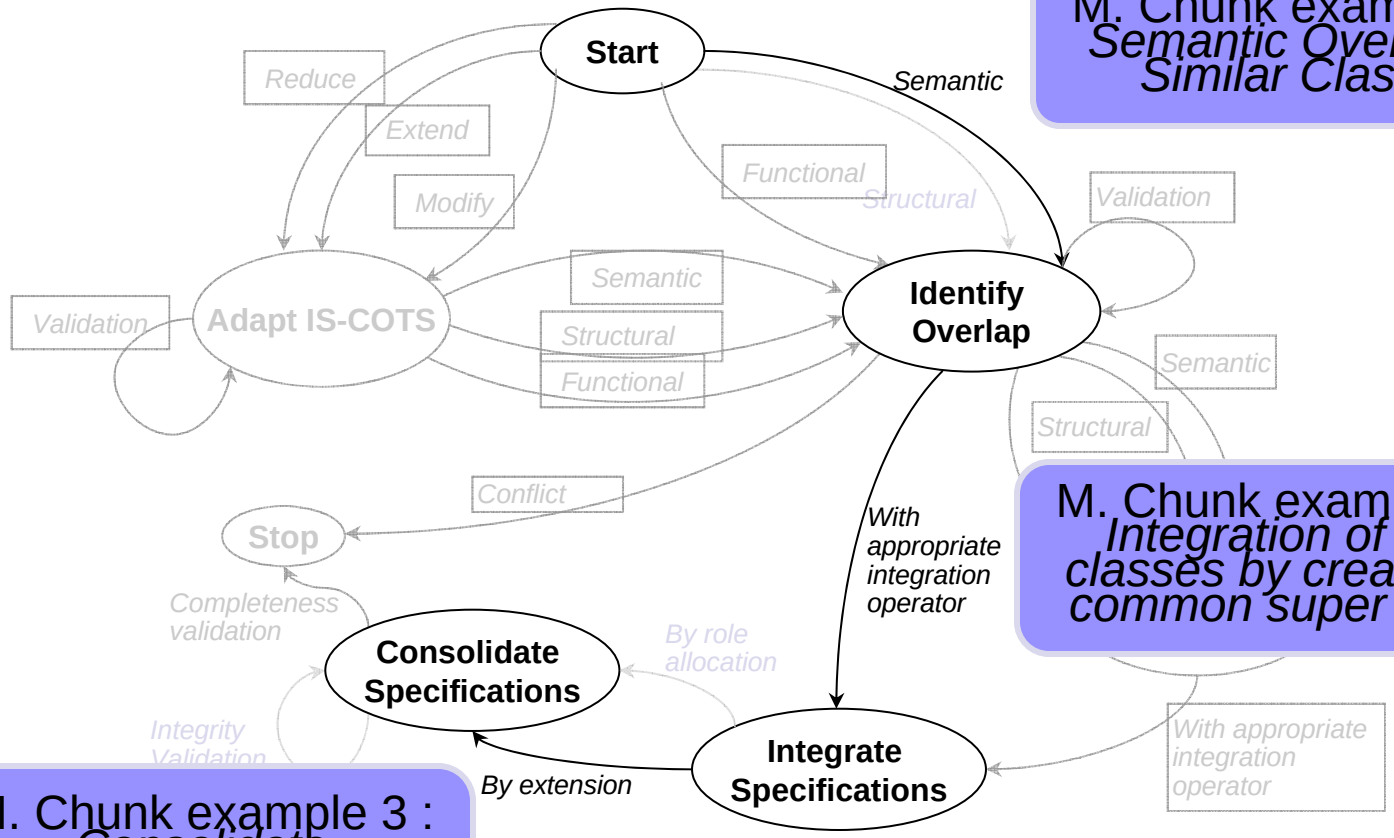
ME approach

1. Adaptation of the selected IS-COTS,
2. Identification of the overlap between the IS-COTS and the IS,
3. Unification of the overlap situations if necessary,
4. Construction of the integrated specification,
5. Consolidation of the obtained specifications.

Requirements MAP



Examples of Method Chunks



M. Chunk example 1 : *Semantic Overlap of Similar Classes*

M. Chunk example 2 : *Integration of two classes by creating a common super class*

M. Chunk example 3 : *Consolidate integration made via superclass creation*

Method chunks

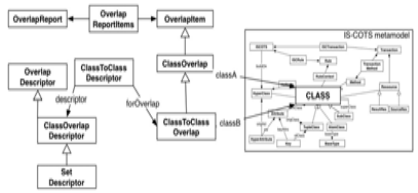
Semantic Overlap of Similar Classes

Situation : Static space specification of the IS and the IS-COTS.

Intension : To produce an overlap report that specifies the semantic relationships between the classes of the IS specification and the similar classes of the IS-COTS specification

Product Part

- The IS-COTS metamodel
- The overlap report model



Process Part

For each classes from the IS that have a semantic relationship with a class of the IS-COTS:

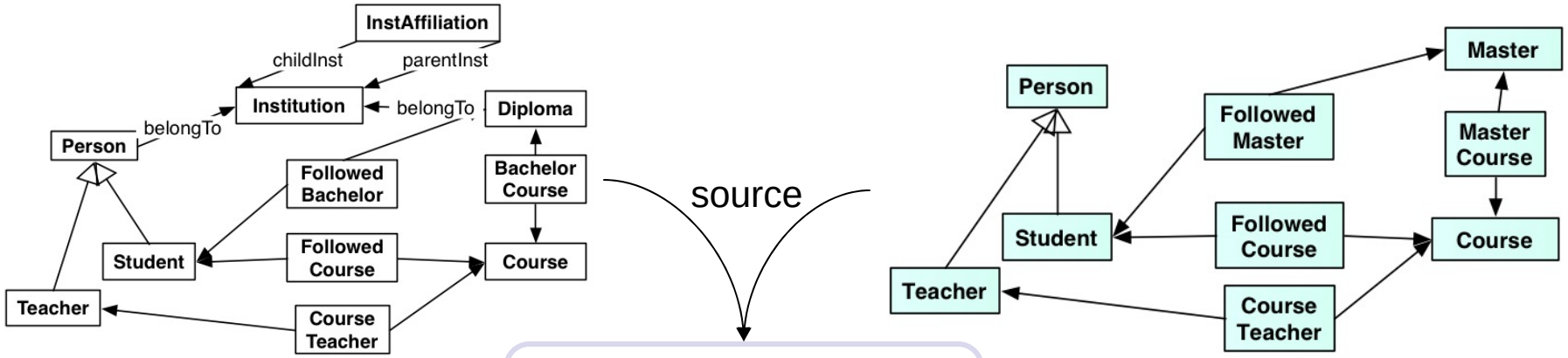
- Create an overlap item in the overlap report,
- Define the relationship descriptor.

Types of relationship descriptor:

1. $**Cl_{is} = **Cl_{is\text{-}cots}$
2. $**Cl_{is} \subset **Cl_{is\text{-}cots}$
3. $**Cl_{is} \supset **Cl_{is\text{-}cot}$
4. $**Cl_{is} \cap **Cl_{is\text{-}cots} \neq \varnothing$
5. $**Cl_{is} \cap **Cl_{is\text{-}cots} = \varnothing$

Case Example$_2$

Applying Chunk N°1



Semantic Overlap of Similar Classes

result

Overlap Report

- **IS.Person = **IS-COTS.Person
- **IS.Teacher = **IS-COTS.Teacher
- **IS.Student ∩ **IS-COTS.Student ≠ ∅
- **IS.Course ∩ **IS-COTS.Course = ∅
- **IS.Diploma ∩ **IS-COTS.Diploma = ∅

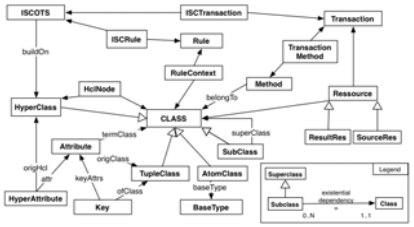
Method chunks

Integration of two Classes by Creating a Common Superclass

Situation : Class Cl_{is} from the IS specification and class $Cl_{is\text{-}cots}$ from the IS-COTS specification.
Intension : To integrate two classes by creating a third class that is a generalization of the two initial ones.

Product Part

The IS-COTS metamodel



Process Part

In the IS

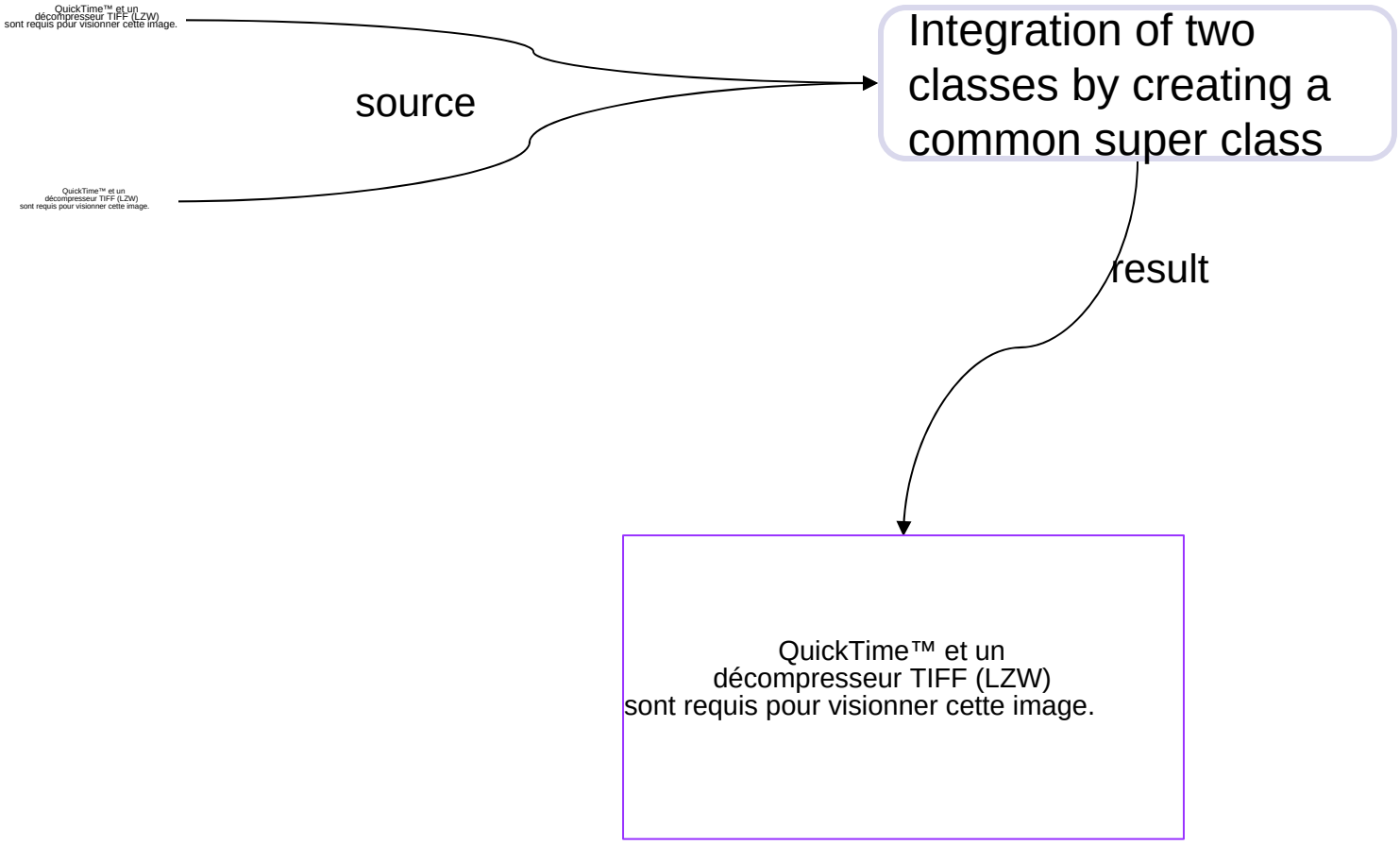
- Add a common class Cl' as a generalization of Clis and Clis-cots.
- Change the IS spec: to make Cl' a superclass of Clsi.
- Move all attributes that are common with Clis-cots from Clis to Cl'.
- Move all methods that are common with Clis-cots from Clis to Cl'.
- Move all transactions that are common with Clis-cots from Clis to Cl'.
- Move all integrity rules that are common with Clis-cots from Clis to Cl'.

In IS-COTS

- Delete from the Clis-cots all attributes that are common with Clis.
- Delete from the Clis-cots all methods that are common with Clis.
- Delete from the Clis-cots all transactions that are common with Clis.
- Delete from the Clis-cots all integrity rules that are common with Clis.
- Change the IS-COTS spec: to make Cl' a superclass of Clis.

Case Example 3

Applying Chunk N°2



Method chunks

Consolidate Integration Made via Superclass Creation

Situation : Class Cl_{is} from the IS specification, class $Cl_{is\text{-}cots}$ from the IS-COTS specification and class Cl' - common super class to Cl_{is} and $Cl_{is\text{-}cots}$.
Intension : To consolidate the integrated specification by adding new transactions and integrity rules.

Product Part

- The IS-COTS metamodel
- The overlap report model

Process Part

If relation descriptor == ($^{**}Cl_{is} \cap {}^{**}Cl_{is\text{-}cots} = \varnothing$) :

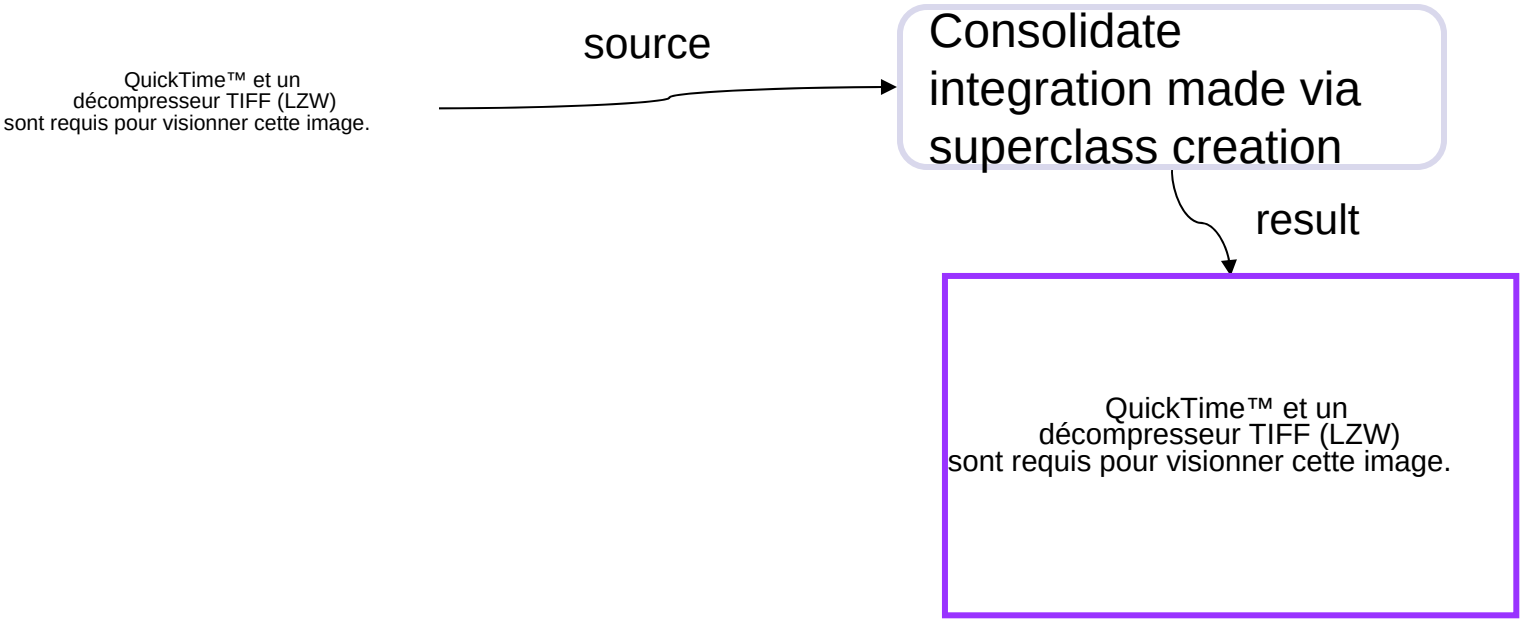
- add a consistency rule ensuring that each object can be only an object of Cl_{is} or of $Cl_{is\text{-}cots}$.

If relation descriptor == ($^{**}Cl_{is} \cap {}^{**}C_{lis\text{-}cots} \neq \varnothing$) :

- determine if a new rule is needed between Cl_{is} and $Cl_{is\text{-}cots}$
- add one or more transactions for objects transfer from one class to the another.

Case Example 4

Applying Chunk N°3



Rule

ON ENTER in MasterStudent : obj exist in
BachelorStudent AND obj.hasBachelor == True
where obj represent a student.

Conclusion

Contributions

- Notion of IS-COTS and its metamodel.
- Requirements specification for a situation-driven approach supporting IS-COTS integration into existing IS.
- Examples of method chunks to be integrated into our approach.

Perspectives

- Identifying and evaluating different situations that can occur in the IS-COTS integration process.
- Defining method chunks satisfying these situations.
- A tool support is also under development.

Questions ?

Thank you for your attention