

Composants de Systèmes d'Information

Slim Turki, Michel Léonard, Nicolas Arni-Bloch

MATIS Geneva Team – Database Laboratory

University of Geneva

<http://matis.unige.ch>



Introduction

- ★ Nouveaux SIs, nouveaux Besoins
 - ★ Évolution
 - ★ Modularité
- ★ Composant de SI
 - ★ Gérer l'ouverture et l'intégration de SIs
 - ★ Environnement offrant un niveau significatif d'abstraction et de modularité
 - ★ responsables de SI :
 - décideurs au niveau de l'information et non seulement au niveau technique

Plan

- ★ Définition d'un SI
 - ★ Espace statique
 - ★ Espace dynamique
 - ★ Règles d'intégrité
- ★ Définition d'un composant de SI
 - ★ Règles de conformité
- ★ Conclusion et perspectives



Systeme d'Information

$SI = \langle ES, ED, RI \rangle$

- ES: espace statique
 - Ensemble de classes et de méthodes
- ED: espace dynamique
 - Cycles de vie
- RI: Règles d'intégrité
 - Préserver la cohérence et la consistance durant l'exploitation



Espace statique

- ✦ Spécification des propriétés statiques d'un SI
- ✦ Diagramme de classe:
 - classes, attributs, clés
 - méthodes des classes
 - dépendances entre classes
 - relations de spécialisation/généralisation
- ✦ Dépendances entre les classes
 - dépendance existentielle
 - Un objet est toujours lié au même objet durant toute sa vie
 - dépendances de spécialisation.
 - Cas particulier des dépendances existentielles

Espace dynamique

- ★ Spécifications dynamiques du SI
 - ★ diagrammes de cycles de vie d'objet
- ★ graphes de Gavroche :
 - ★ graphes bipartis / réseaux nœuds-étoiles
 - ★ les nœuds correspondent un à un à des classes
 - ★ et les étoiles à des transactions.



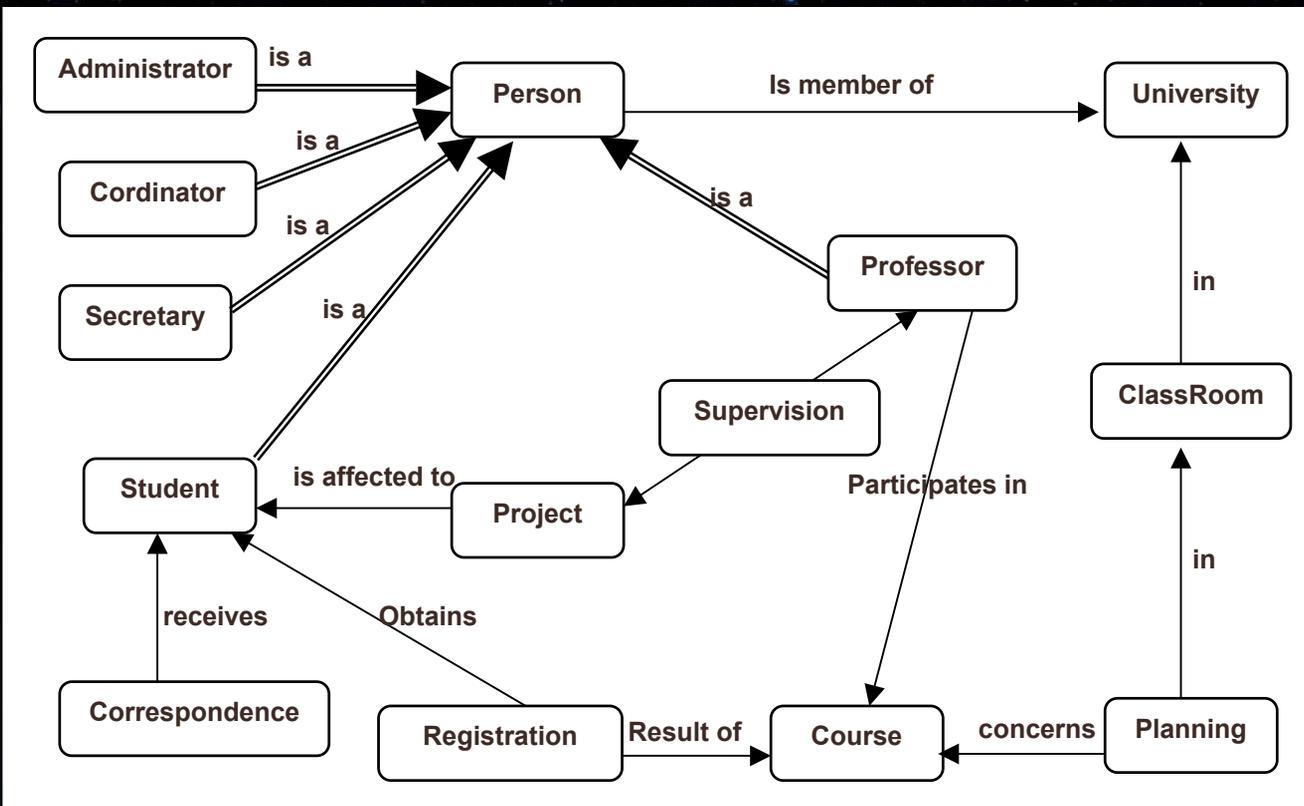
Transaction

★ Une transaction tr

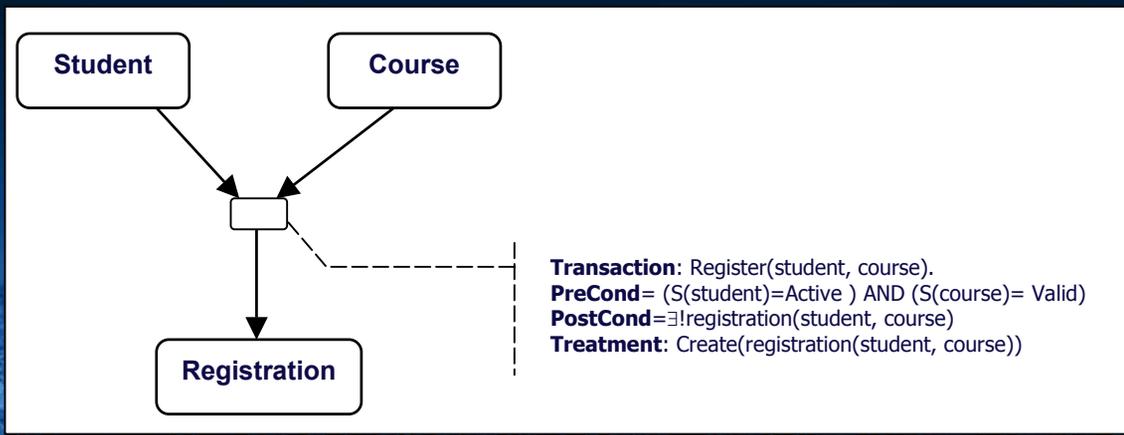
- ★ une ou plusieurs classes (nœuds) de base (${}^{\circ}tr$) en entrée
- ★ une ou plusieurs classes de rajout comme résultat (tr°).
- ★ tr utilise les informations contenues dans les objets des nœuds ${}^{\circ}tr$ et produit des objets des classes tr° .
- ★ Conditions en entrée et en sortie :
 - peuvent être des expressions booléennes complexes.

★ domaine de tr :

- union des ses classes en entrée et ses classes en sortie.
- $Dtr = {}^{\circ}tr \cup tr^{\circ}$



M@TIS Informational Schema



Règles d'intégrité

★ RIs

- ★ constituante essentielle d'un SI.
- ★ préserver sa cohérence et sa consistance durant son exploitation.
- ★ condition définie sur une ou plusieurs classes.

★ Le contexte d'une RI

- ★ Ensemble des classes concernées par sa définition et sa validation.

Composant de SI

- ☀ CSI= SI autonome

$$CSI = \langle ESc, EDc, RIc \rangle$$

- ESc: espace statique du CSI
- EDc: espace dynamique du CSI
- RIc: Règles d'intégrité du CSI

- ☀ Ensemble de conditions pour garantir son autonomie et sa cohérence



Règles de conformité

$CSI = \langle ESC, EDC, RIC \rangle$

1- Unité sémantique

2- Complétude des classes :

- si une classe cl_i appartient à un CSI et référence une classe cl_j , alors cl_j appartient au CSI.

3- Cohérence des transactions :

- toutes les transactions du CSI valident toutes les RIs du CSI.

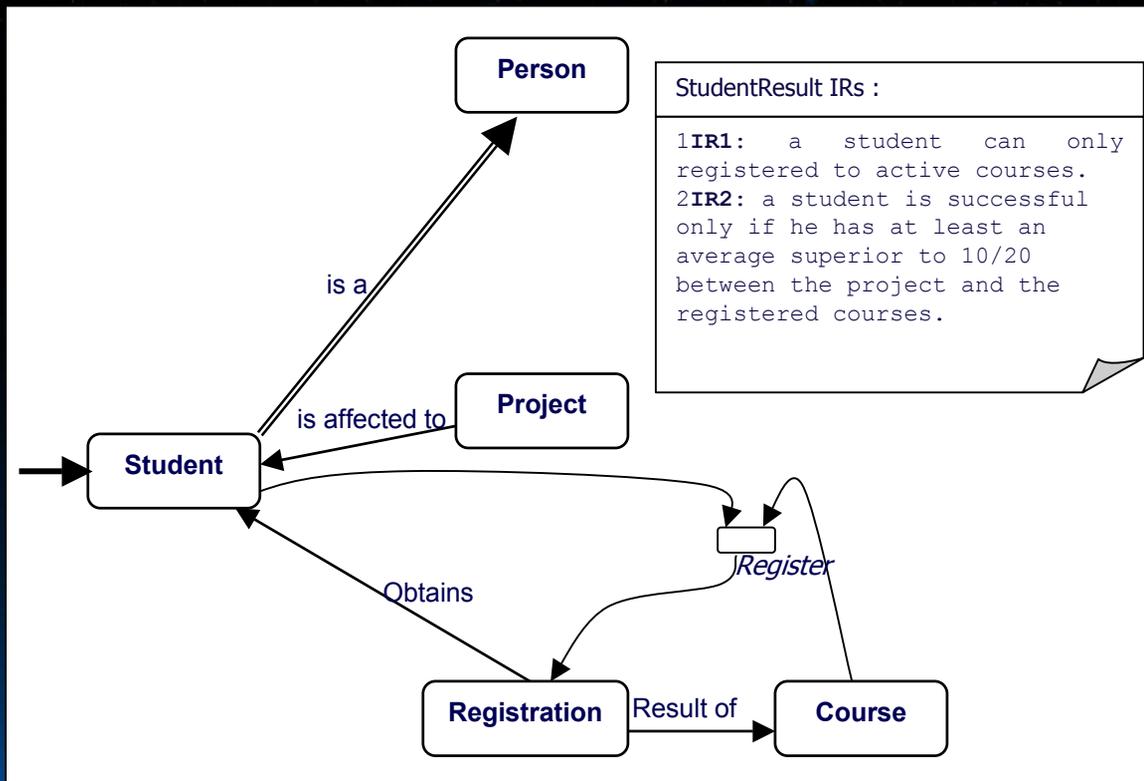
4- Complétude des transactions :

- si une transaction tr appartient au CSI, toutes les classes appartenant au domaine de tr appartiennent au CSI.

5- Complétude des RIs :

- toutes les classes appartenant au contexte d'une RI appartiennent au CSI.

Exemple de composant



Manipulation des CSI

- ★ Implémentation des opérations de définition et de manipulation pour adapter, raffiner ou intégrer un CSI:

- [Créer]

- [Extraire]

- [Modifier]

- [Intégrer] [Specialiser/generaliser] [Condenser] [Développer] [Etudier les parties communes à 2 CSIs] ...

Conclusion

- ✦ Créer un environnement permettant de gérer les CSIs
- ✦ Concept de CSI:
 - ✦ Générique, non dédié à une technologie, une méthode ou un SI particulier
 - ✦ Pratiquement mis en œuvre pour le développement et l'adaptation de SIs académiques.

Composants de Systèmes d'Information

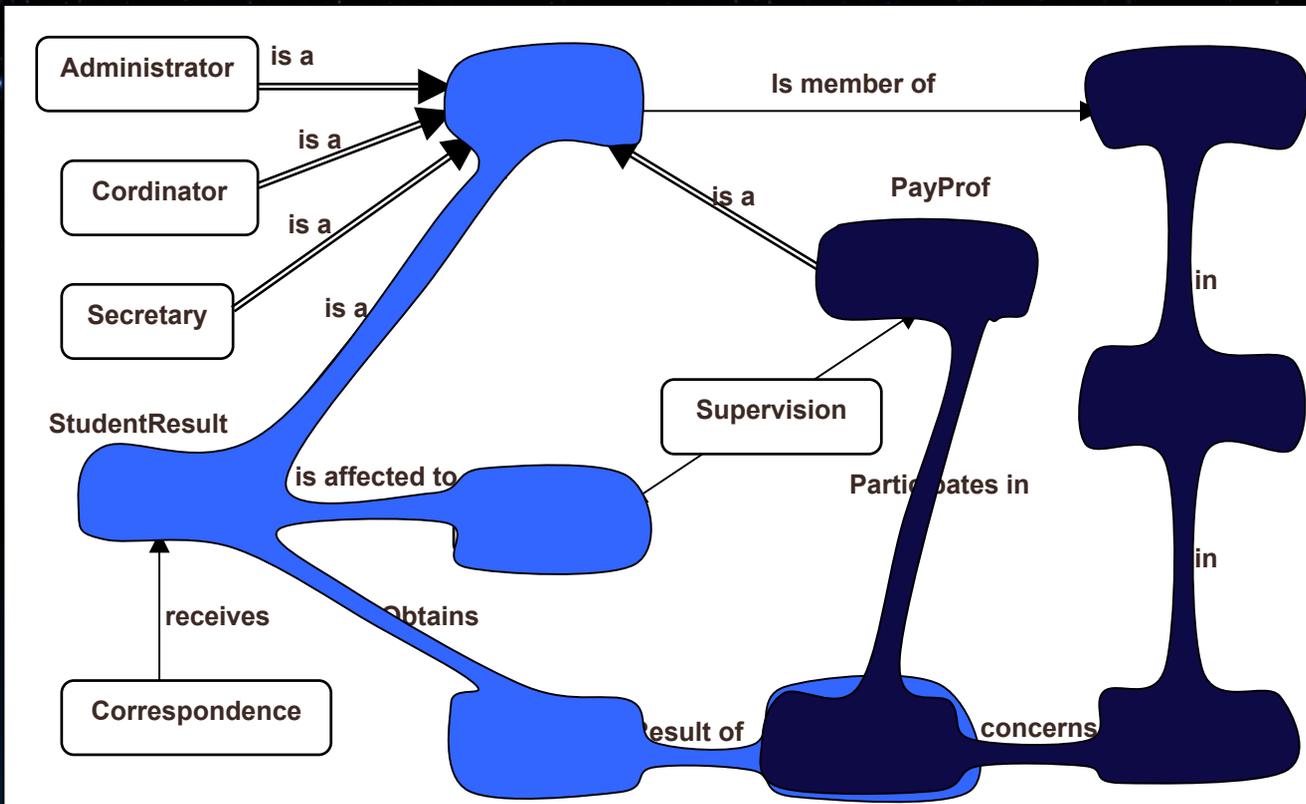
Slim Turki, Michel Léonard, Nicolas Arni-Bloch

MATIS Geneva Team – Database Laboratory

University of Geneva

<http://matis.unige.ch>





- : An object o_A of A is linked to at most one object o_B of B. o_B is inversely linked to o_A and eventually to other objects of A.
- : A is a subclass of B