

SUJET

Réalisation d 'un éditeur permettant de programmer graphiquement (UML étendu) une application conformément au modèle OFL

Étudiants

Alonzo Laurent
Colletin Guillaume
Garoste Fanny

Encadreurs

Crescenzo Pierre
Lahire Philippe

PLAN

- I Présentation du contexte
- II Méta Langage
- III Découverte du modèle OFL
- IV Description du sujet
- V Détails techniques
- VI Conclusion
- VII Démonstration

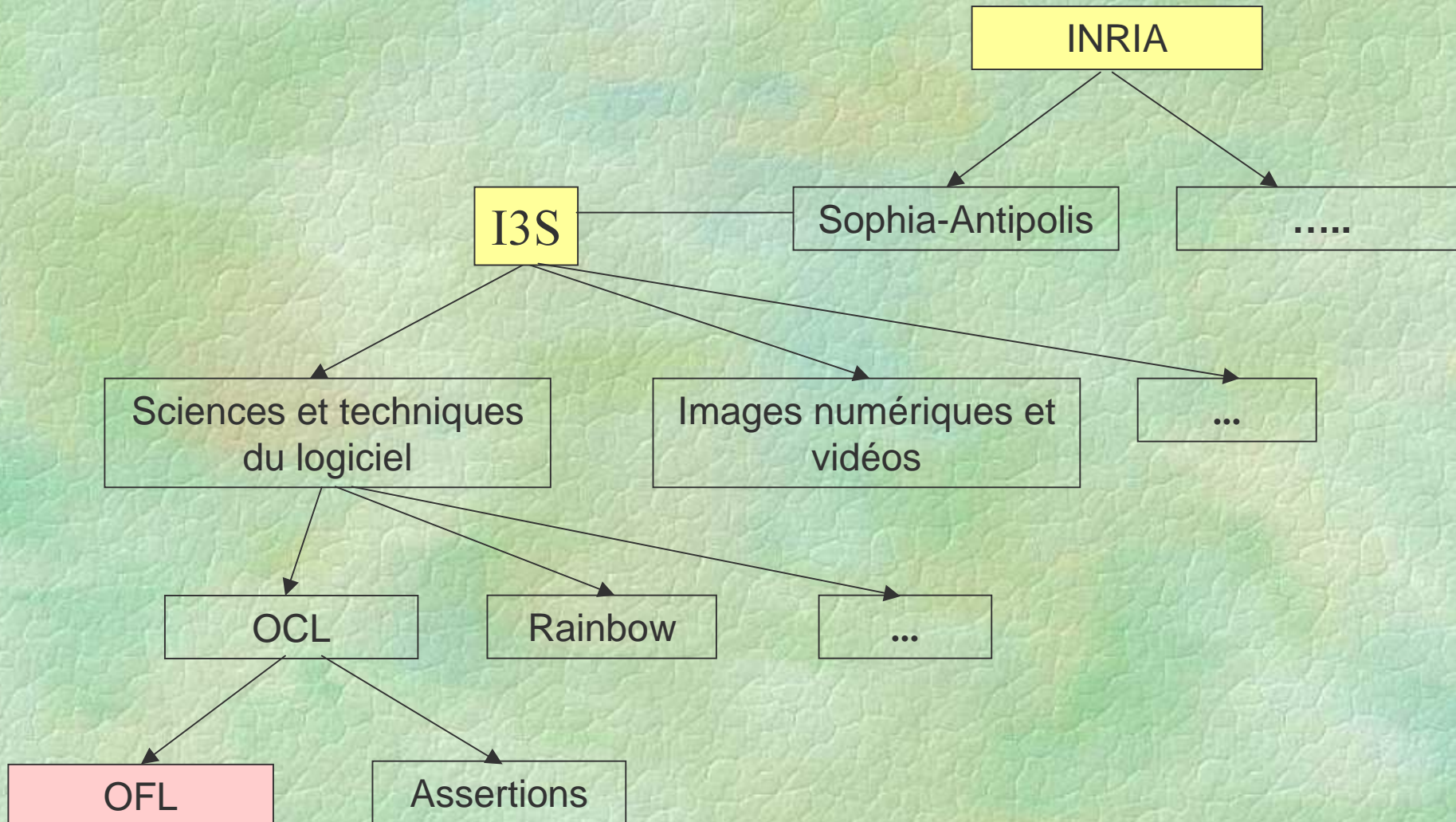
Présentation du contexte

1/2

- Projet élaboré au sein du laboratoire I3S
- 3 équipes au sein de I3S fonctionnant en partenariat avec l'unité de recherche Sophia-Antipolis de l'INRIA
- OFL : opération de recherche du projet OCL dans le thème « Sciences et techniques du logiciel »

Présentation du contexte

2/2



Méta langage

1/3

- Permet de définir un langage de programmation
- Créer son propre langage adapté à ses besoins
- Modifier un langage existant

Méta langage

2/3

Objets manipulés en méta programmation :

instances de classes représentant :

- un type de description (interface, classe)
- un type de relation (agrégation, héritage)
- tout autre type d'objet décrivant le comportement et la sémantique opérationnelle d'un langage.

Méta langage

3/3

Niveau méta langage

Concepts
(Méta composants)

Description

Relation

Niveau langage

Composants
(Méta objets)

Classe

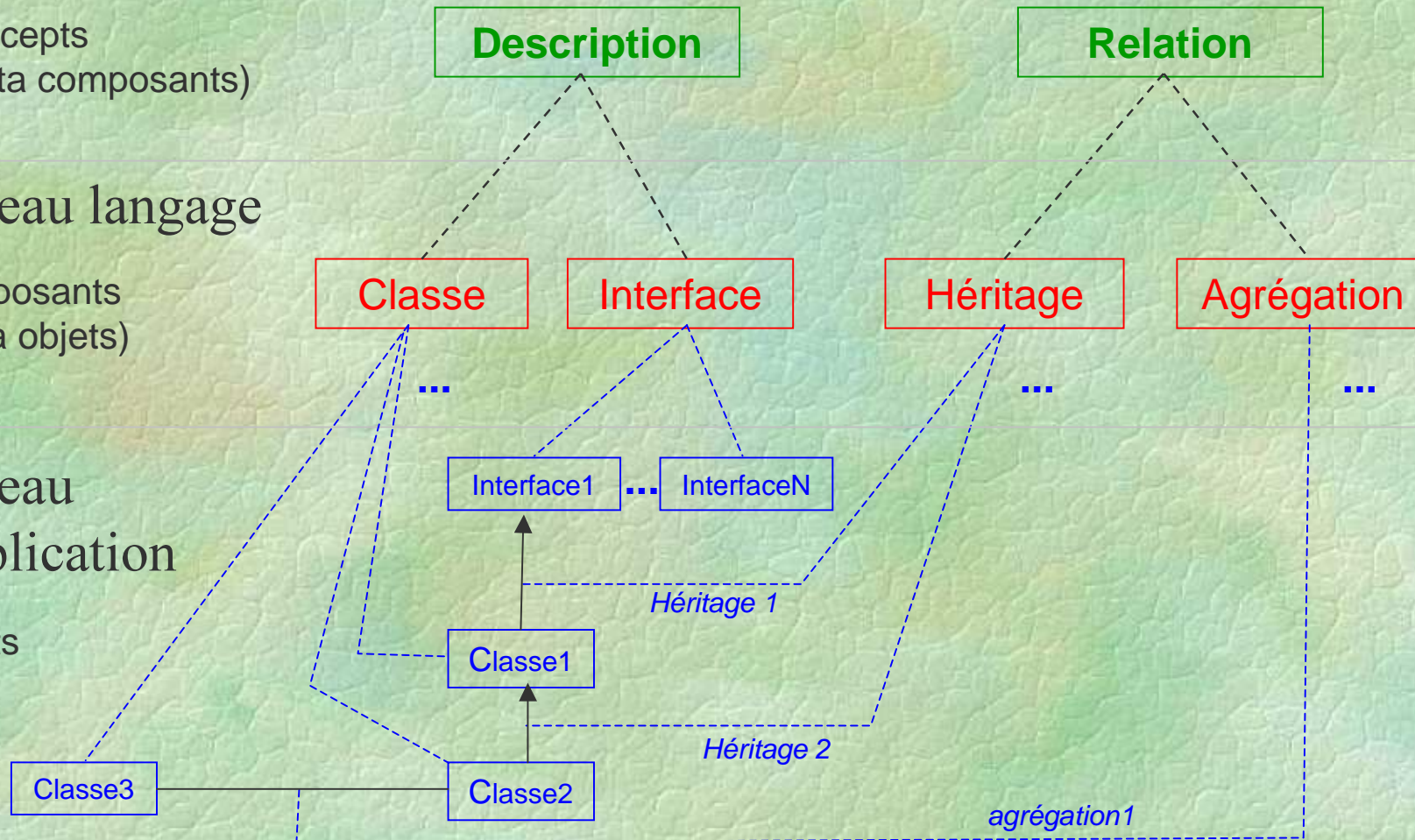
Interface

Héritage

Agrégation

Niveau Application

Objets



Découverte du modèle OFL 1/2

C'est un méta langage objets

- il est indépendant d'un langage particulier
- il regroupe tous les composants des langages objets existants
- il permet de définir un langage avec un comportement particulier

Découverte du modèle OFL 2/2

Trois niveaux :

- Le niveau « OFL »
- Le niveau « LANGAGE »
- Le niveau « APPLICATION »

Description du sujet 1/2

Proposer une interface graphique permettant de développer une application à partir d'un langage.

L'utilisateur doit pouvoir :

- récupérer un langage existant
- créer une description (liste de méta-descriptions)
- créer une relation (liste de méta-relations)

Description du sujet 2/2

L'interface doit permettre à l'utilisateur :

- d'accéder à l'ensemble des composants (arbre)
- de visualiser l'ensemble des descriptions déjà créées (arbre)
- de créer et de manipuler des descriptions et des relations (zone graphique)
- d'utiliser les options du logiciel (menu contextuel)

Détails techniques

1/3

Analyse du comportement d'une description :
paramètres du Component-Description référencé

- Name (ex : class, interface)
- Attribute (ex : allowed | forbidden)
- Method (ex : allowed | forbidden)
- Genericity (ex : true | false)

Détails techniques

2/3

Analyse du comportement d'une relation :
paramètres du Component-Relationship référencé

- Name (ex : spécialisation)
- Kind (ex : import | use)
- Cardinality (ex : $\langle 1, 1 \rangle$ | $\langle 2, 3 \rangle$)
- Symmetry (ex : true | false)

Détails techniques

3/3

Analyse du comportement d'un langage :
paramètres du Component-Language référencé

- Name (ex : monLangage)
- Valid description-components
- Valid relationship-components
- Valid relationships

Conclusion

- Utilisation d'un langage personnalisé et approprié
- Développement facile et rapide d'une application
- Création d'une application indépendante d'un langage donné

- => Le modèle OFL permet donc à l'utilisateur de définir le langage le plus adapté afin de développer son application

Démonstration ...

