



Institut Mines-Télécom

Evolution d'architectures logicielles à base de composants

Alexandre Le Borgne

Directeurs :

- David Delahaye (LIRMM)
- Marianne Huchard (LIRMM)

Encadrants :

- Christelle Urtado
- Sylvain Vauttier





Plan

■ Présentation du sujet

- Définitions
- Architecture Description Language
- Evolution
- Résumé du sujet
- Prélude d'état de l'art

■ Pistes & Interrogations

- Versionnement
- Interrogations
- Orientations

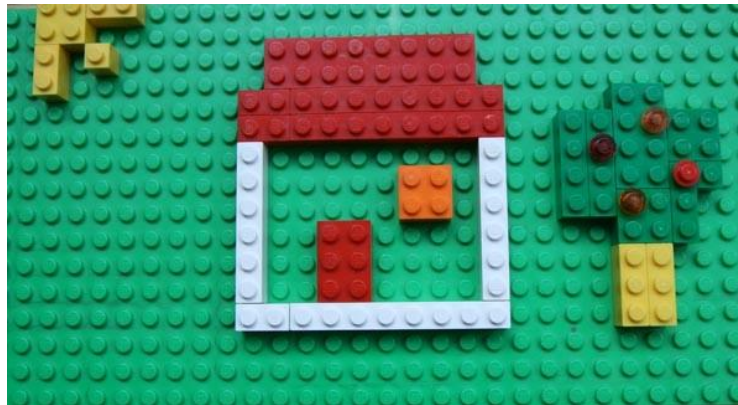
Définitions

■ Composant

- *“A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to third-party composition.” Szyperski*



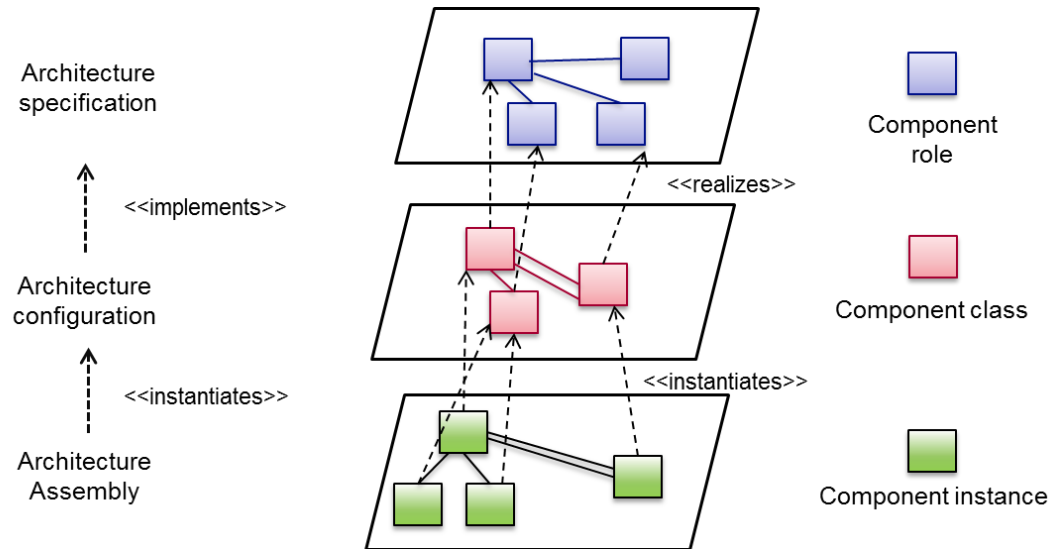
■ Architecture



Architecture Description Language

■ Dedal

- Issu de la thèse d' Huaxi (Yulin) Zhang
- 3 niveaux correspondant aux principales étapes de développement
 - Approche de l'Ingénierie Des Modèles
 - Gestion de développement d'architectures logicielles à base de composants

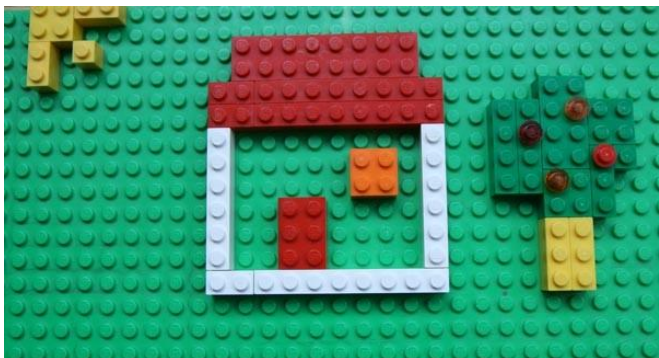


Evolution

- **Architectures logicielles en constante évolution**
- **Traité dans la thèse d'Abderraman Mokni**
 - Evolution automatique

- **Evolution dans Dedal**
 - **Besoin d'une gestion rigoureuse de cette évolution**
 - Préserver de la traçabilité
 - Eviter :
 - inconsistances
 - Incohérences
 - pertes de décisions architecturales
 - **Formalisation des concepts de Dedal**
 - Langage B
 - **Proposer automatiquement un plan d'évolution**
 - Coévolution
 - Dérive
 - Erosion
 - Propagation des changements dans les 3 niveaux d'architecture

Evolution



Plan d'évolution



But



Mon sujet

■ Sémantique des relations

- Etudier les relations qui existent entre le trois niveaux d'architecture de Dedal
- Renforcer la définition formelle de ces relations

■ Versionnement

- Gestion de versions de composants dans des librairies de composants
- Définir le versionnement dans Dedal

Prélude d'état de l'art – Versionnement

- **STUCKENHOLZ, Alexander. Component evolution and versioning state of the art. *ACM SIGSOFT Software Engineering Notes*, 2005, vol. 30, no 1, p. 7.**

	Oberon, Component Pascal	Unix Library Versioning	CORBA, Web-Services	Windows XP	.NET Components	Java Versioning	McCamant and Ernst	SOFA
versioning support	○	●	●	●	●			●
detection of changes	●					●	●	
substitutability checks						●	●	
minimizing incompatibilities								

○ basic support, ● full support

Table 5.1: Comparison of component models, programming languages and frameworks

- Challenges
 - Vérification de la compatibilité
 - Réduire les conflits de version
 - Version reachability graph

Prélude d'état de l'art – Versionnement

- **BAUML, Jaroslav et BRADA, Premek. Automated versioning in OSGi: A mechanism for component software consistency guarantee. In : 2009 35th Euromicro Conference on Software Engineering and Advanced Applications. IEEE, 2009. p. 428-435.**
 - Problème
 - Fiabilité des identificateurs de version
 - Tous les composants doivent adhérer à la même convention de versionnement
 - Amélioration de la sémantique des identificateurs
 - Génération automatique de ces identificateurs

Prélude d'état de l'art – Versionnement

- **PAIGE, Richard F., MATRAGKAS, Nicholas, et ROSE, Louis M. Evolving models in model-driven engineering: State-of-the-art and future challenges. *Journal of Systems and Software*, 2016, vol. 111, p. 272-280.**
 - Coévolution
 - modèle / méta-modèle
 - Opération / méta-modèle
 - Versionnement de modèle
 - Challenges
 - Scalability
 - Automatisation
 - Hétérogénéité des dépendances

Problématiques du versionnement

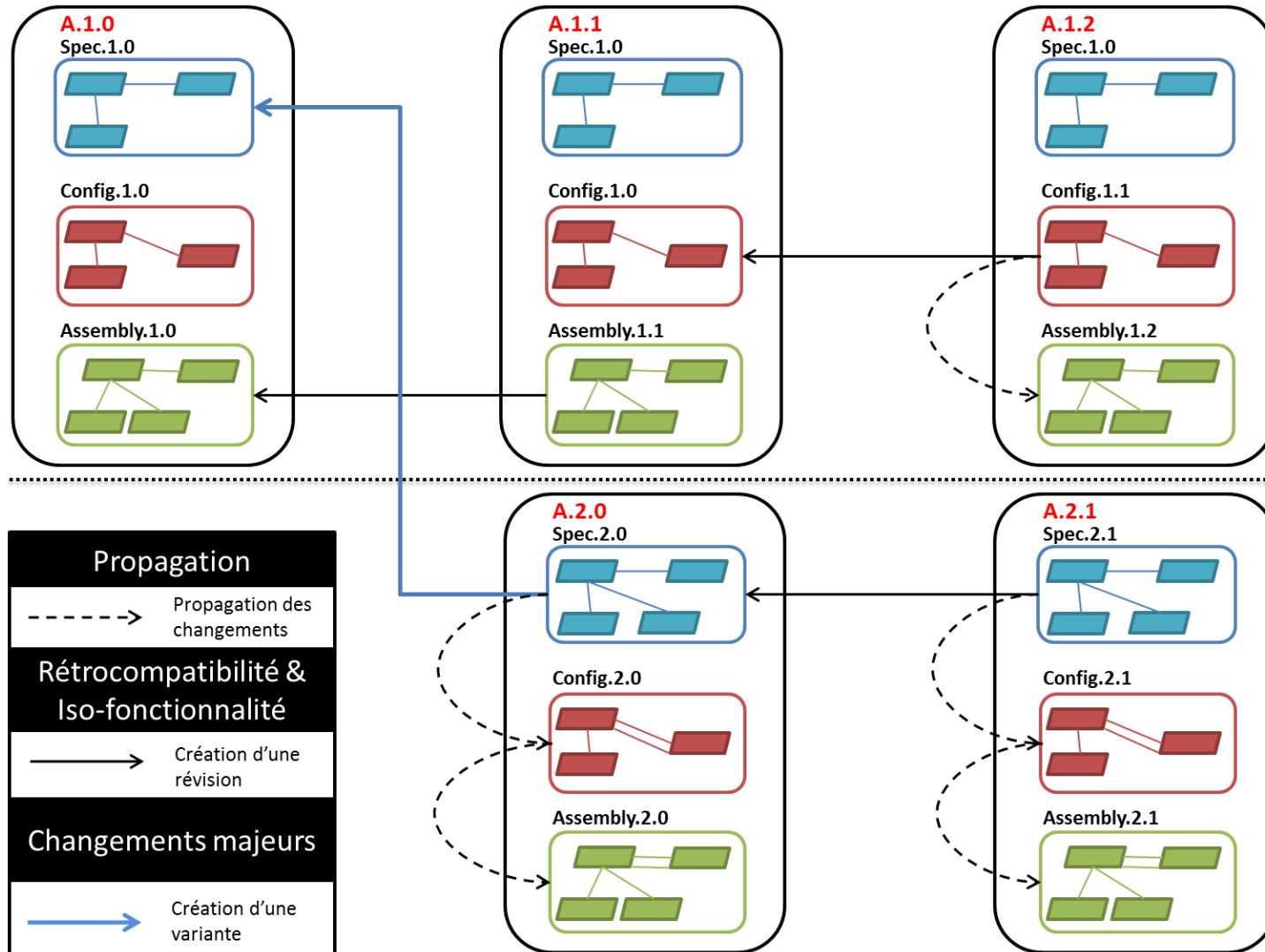
■ Versionnement multi-niveaux (Dedal)

- Spécification
- Configuration
- Assemblage

■ Automatiser le versionnement

- Détection des changements
 - Types de changements
 - Calcul de compatibilité
 - Guidage de l'architecte (intention) & classement automatique
 - Révision
 - **Rétrocompatibilité**
 - **Iso-fonctionnalité**
 - Variante

Versionnement d'architectures Dedal



Interrogations

- **Problématique de compatibilité dans le versionnement.**
 - Pourrait-on utiliser les propriétés du raffinement de B pour prouver la compatibilité de deux versions d'un composant?
 - Peut-on considérer une révision d'un composant comme un raffinement de ce dit composant?

- **Le raffinement de B ou de Event-B permettent-ils de formaliser les liens entre les différents niveaux d'architecture?**

Orientations

- **Formaliser les relations qui existent dans Dedal**
 - Relations inter-niveaux

- **Renforcer le métamodèle de Dedal**
 - Lien description formelle – métamodèle EMF

- **Formaliser les relations de versionnement**
 - Détecter les types de changements
 - Révision
 - Détecter la **rétrocompatibilité** et l'**iso-fonctionnalité**
 - Variante
 - Propager le changement selon des stratégies à établir
 - Coévolution d'architecture multi-niveaux
 - Automatiser le versionnement



Orientations

- **Développement d'un démonstrateur**
 - Outil intégré à eclipse
 - Guider l'architecte dans le versionnement
 - Post-changement
 - Intention
 - Distribuer une suite de développement



Merci

