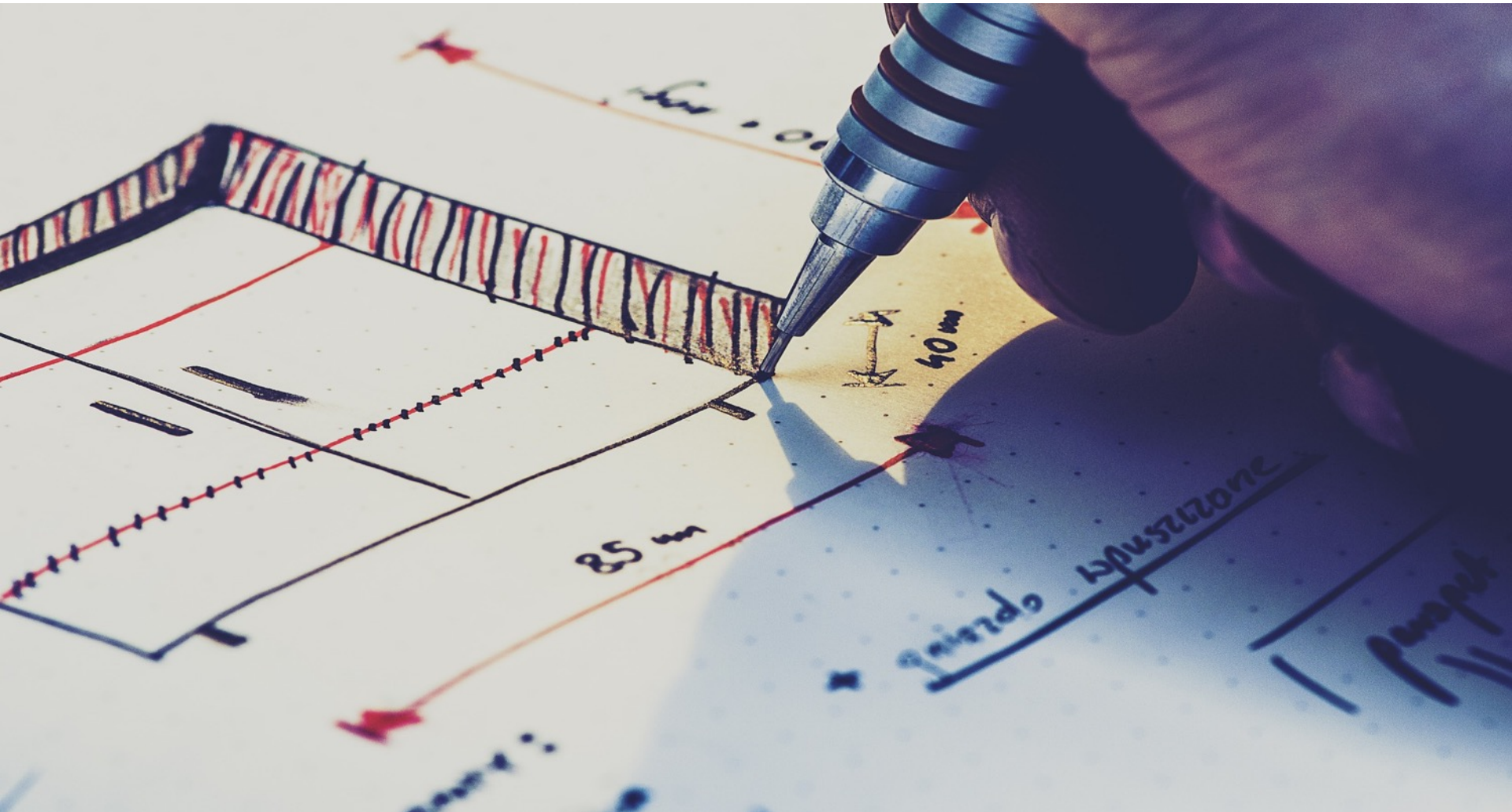


Conception logicielle



Objectifs



- Savoir analyser et concevoir « objet »
- Connaître la notation UML
- Connaître les limites
- Connaître les patrons de conception les plus courants
- Savoir quand les appliquer et ne pas les appliquer

Contrôle des connaissances

- Note (individualisée) de TD long en équipes : 20 %
- Note (individualisée) de projet en équipes : 40 %
- Examen écrit final : 40 %
- Absences injustifiées en TD ? Baisse des notes individualisées de TD long et/ou projet

Communication

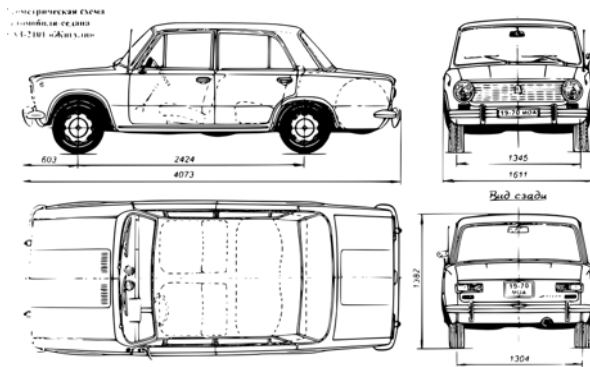
- Diffusion des supports, sujets, etc. :
<https://wiki.unice.fr/display/~collet/Conception+logicielle++-+SI4+-+2019-20>
- Canal par défaut : **SLACK #si4-conception**
- Question sur un des sujets de TD/projet : sur la chaine, pas de mp
 - Tout le monde profite de la réponse
 - Vous vous répondez entre vous

Formation des équipes

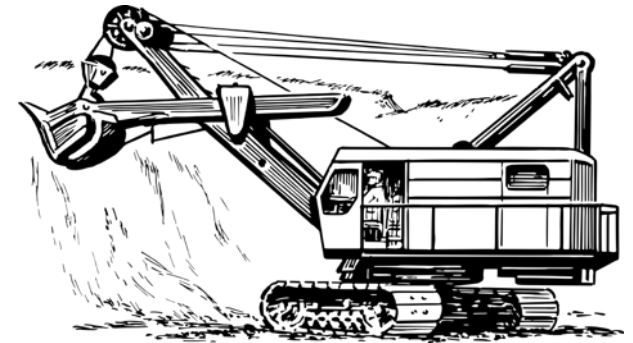
- 4 personnes par équipe (3 sinon)
 - 2 personnes au moins de 2 provenances différentes en entrée de 3A
 - Possibilité d'être dans des TDs différents
- Liste des étudiants de l'équipe à publier sur le slack (chaîne #si4-conception) avant ce soir 22h (mardi 17 septembre !)
 - Nom prénom pour chacun
 - 1 par ligne

Pourquoi modéliser ?

Spécifier la structure et le comportement d'un système



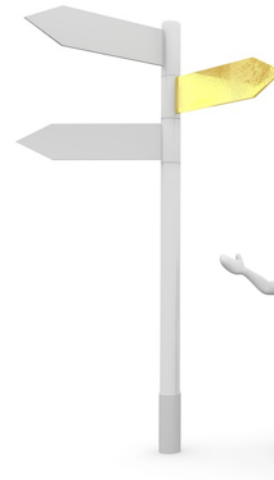
Aider à la construction d'un système



Visualiser un système



Documenter les décisions



UML un peu d'histoire



- Plein de notations et de méthodes différentes à la fin des années 90 : OMT, Booch, etc.
- Besoin d'un langage « standard »
- Plusieurs auteurs de méthodes orientées objets s'allient et créent l'entreprise Rational
 - Grady Booch, Ivar Jacobson, James Rumbaugh

UML, avec un U comme Unified

UML : un mélange de plusieurs notations

- Développé par Rational puis standardisé par l'OMG (Object Management Group) à partir de 1997
- Tous les grandes compagnies entrent dans l'OMG : Oracle, HP, Microsoft, IBM...

Dernière version :
UML 2.5.1,
Dec. 2017 –
800 pages de
spécification...



UML : standard industriel !

- En 2005 déjà, 97% des développeurs connaissaient UML et 56% l'utilisaient dans leurs projets
 - <http://www.prweb.com/releases/2005/04/prweb231386.htm>
- Mais c'est quoi ? Une notation visuelle...
 - Descriptions graphiques et textuelles
 - Syntaxe et sémantique (presque pas ambiguë)
 - Architecture et comportement
 - Génération ou rétro-ingénierie de code

UML : des points forts



- Un langage normalisé
 - Précision
 - Stabilité
 - Facilite l'outillage
- Un support de communication
 - Cadrage de l'analyse
 - Compréhension d'abstractions complexes
 - Universalité

UML : des points faibles



- Période d'adaptation nécessaire
- Lourdeur de la spécification
- UML n'est qu'une notation, l'utiliser ne donne pas de « méthode » pour bien concevoir

Organisation d'UML

- 13 diagrammes réalisés à partir des besoins utilisateurs
- Des aspects fonctionnels
- Des aspects liés à l'architecture

Les vues d'un système

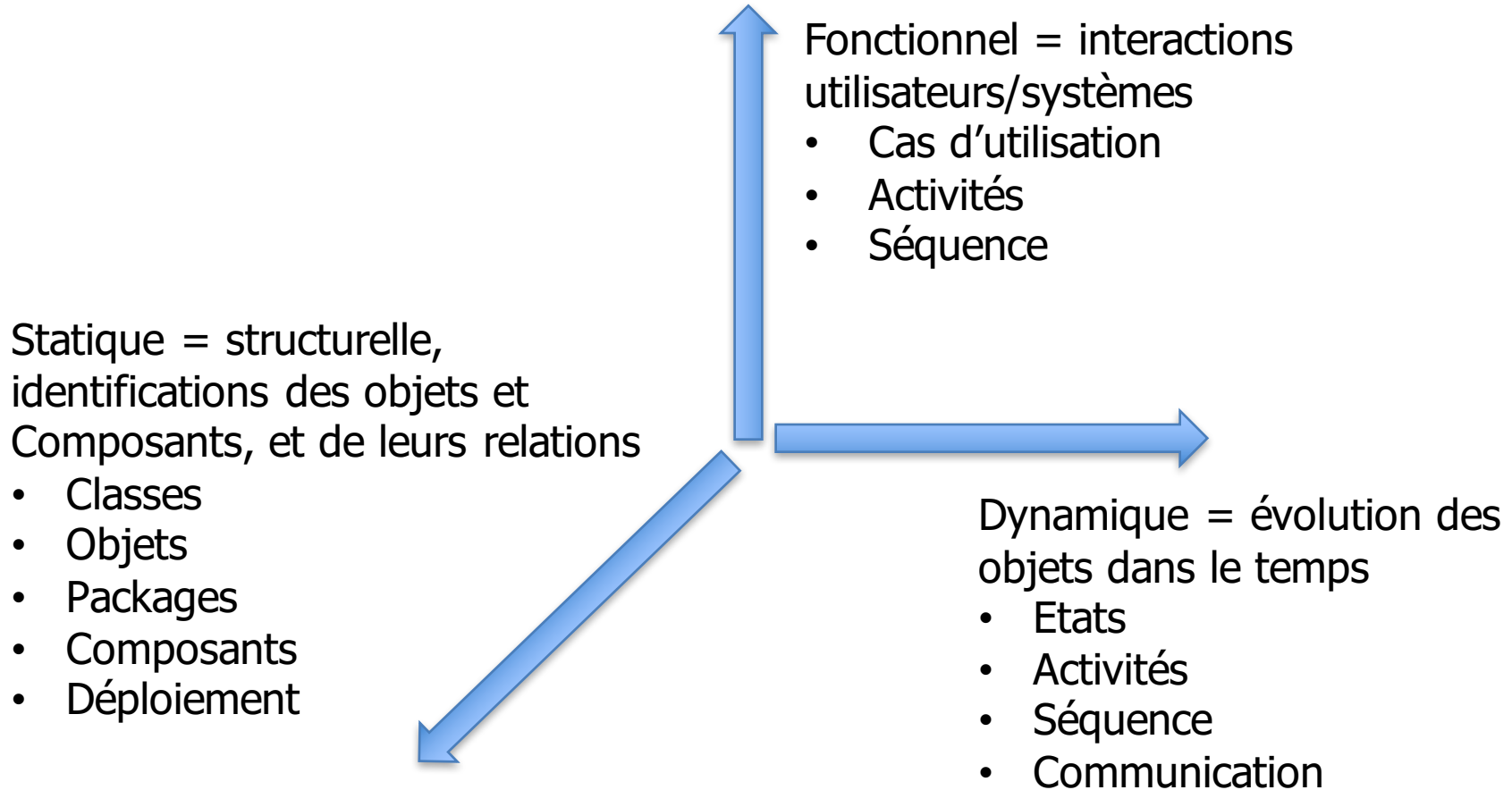


Diagramme de cas d'utilisation

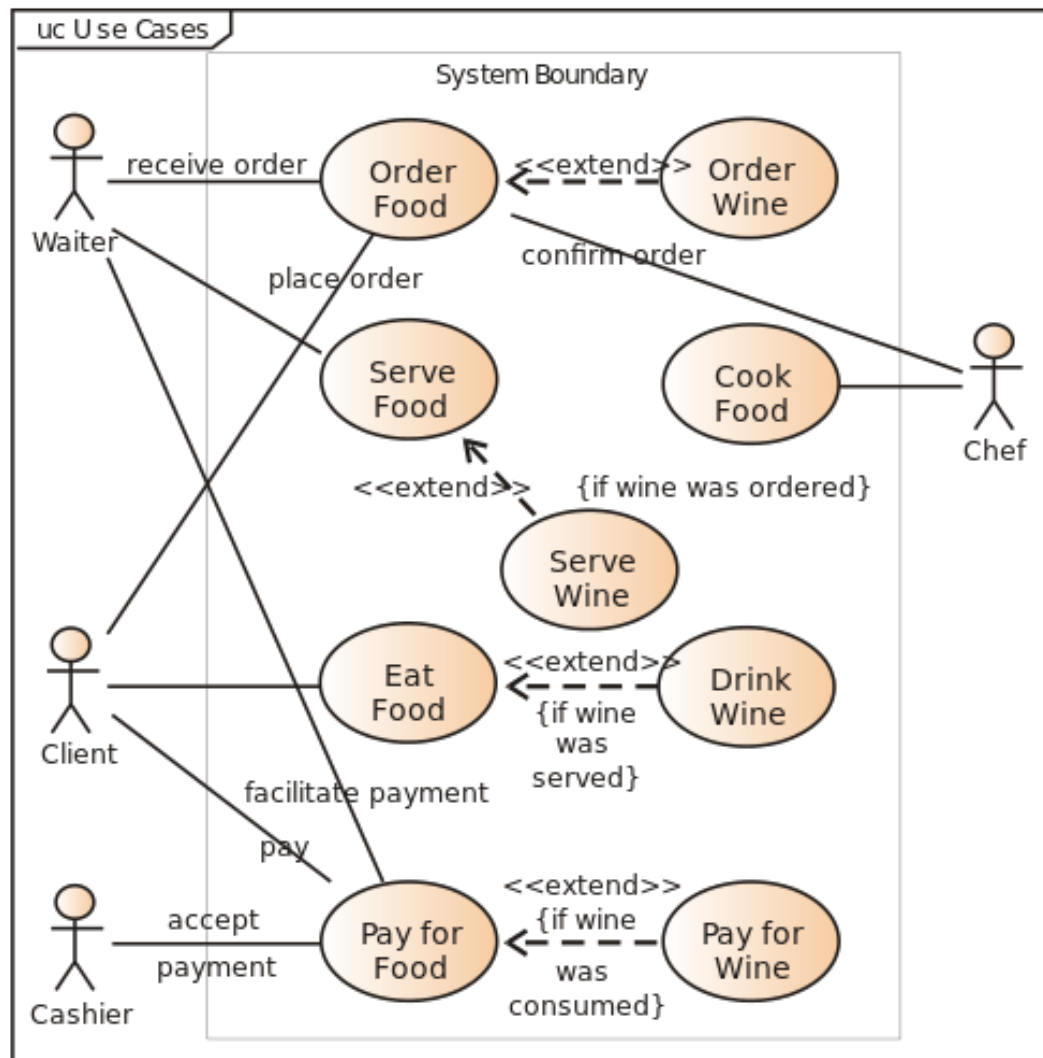
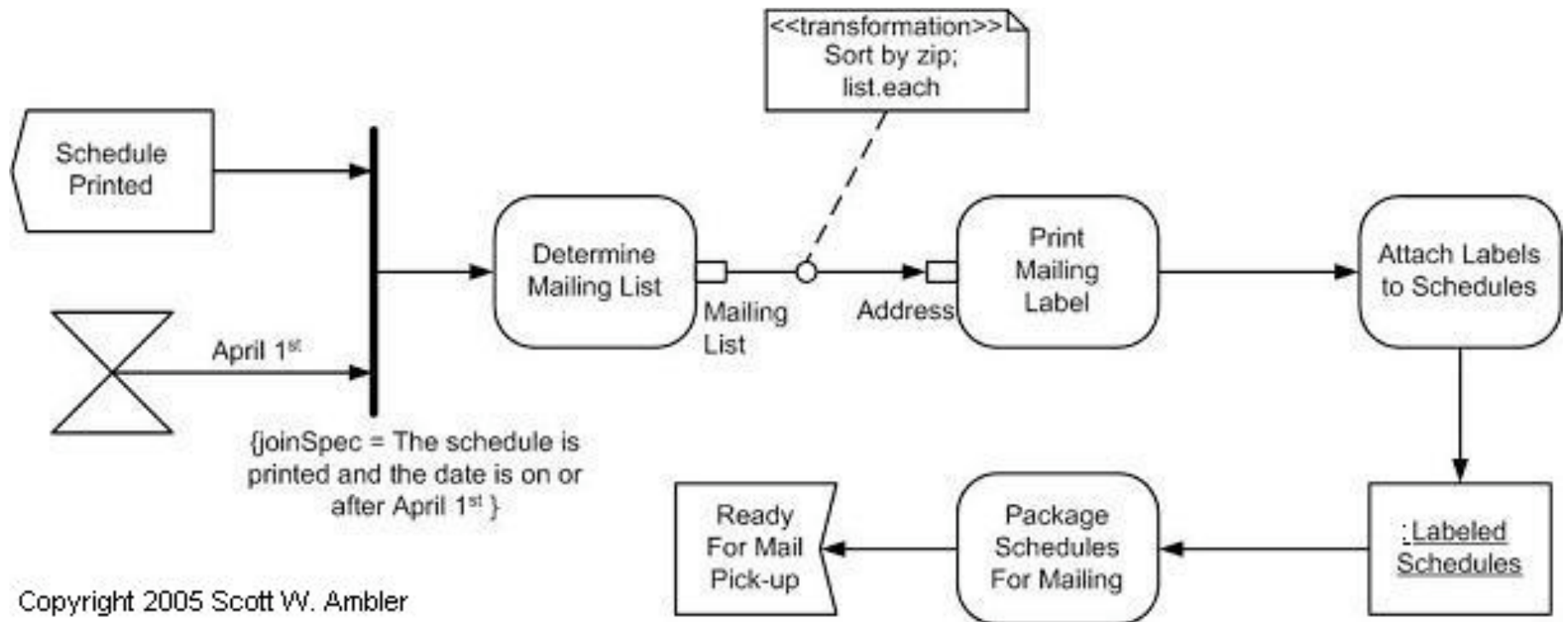


Diagramme d'activités



Copyright 2005 Scott W. Ambler

Diagramme de séquences

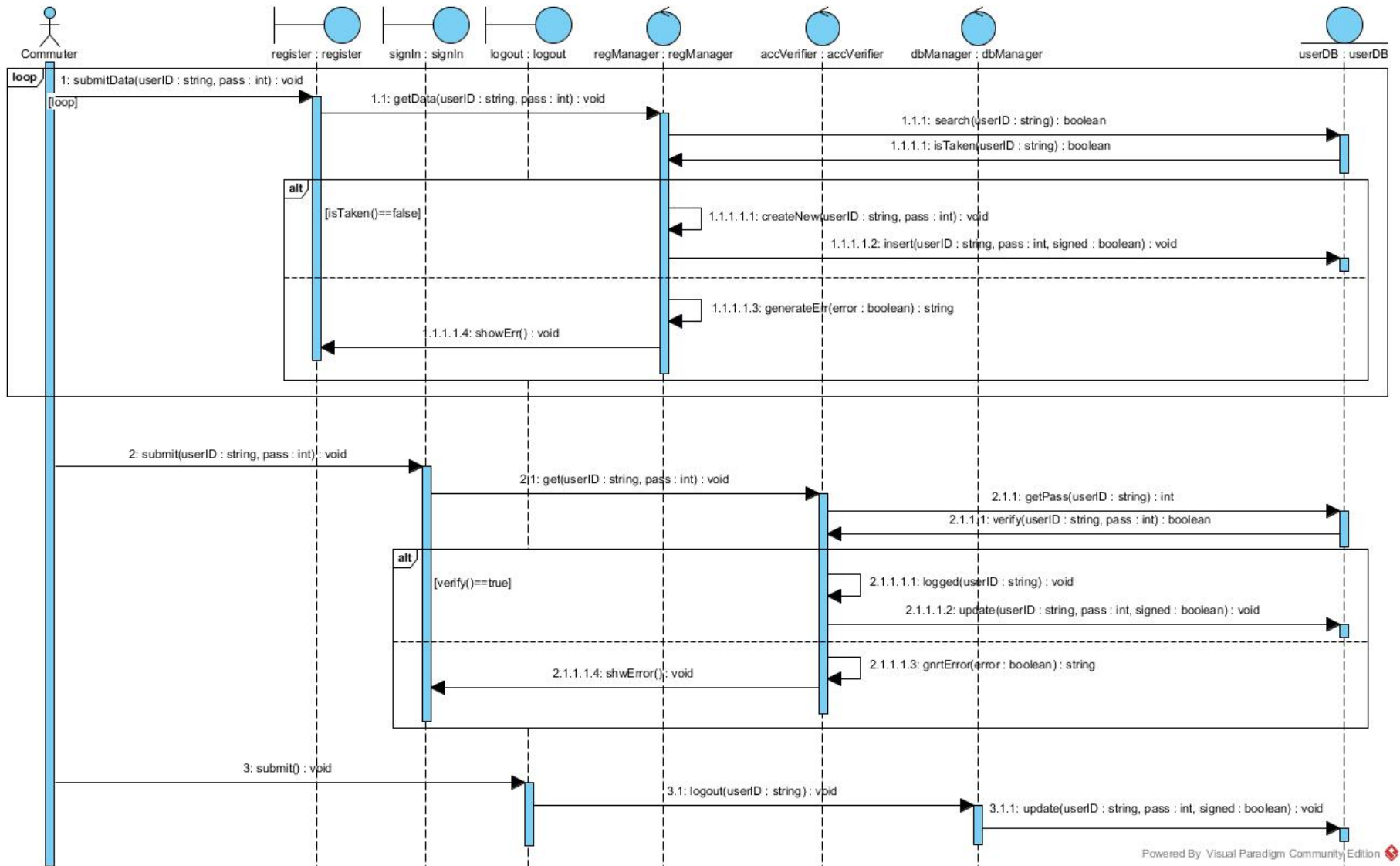


Diagramme de classes

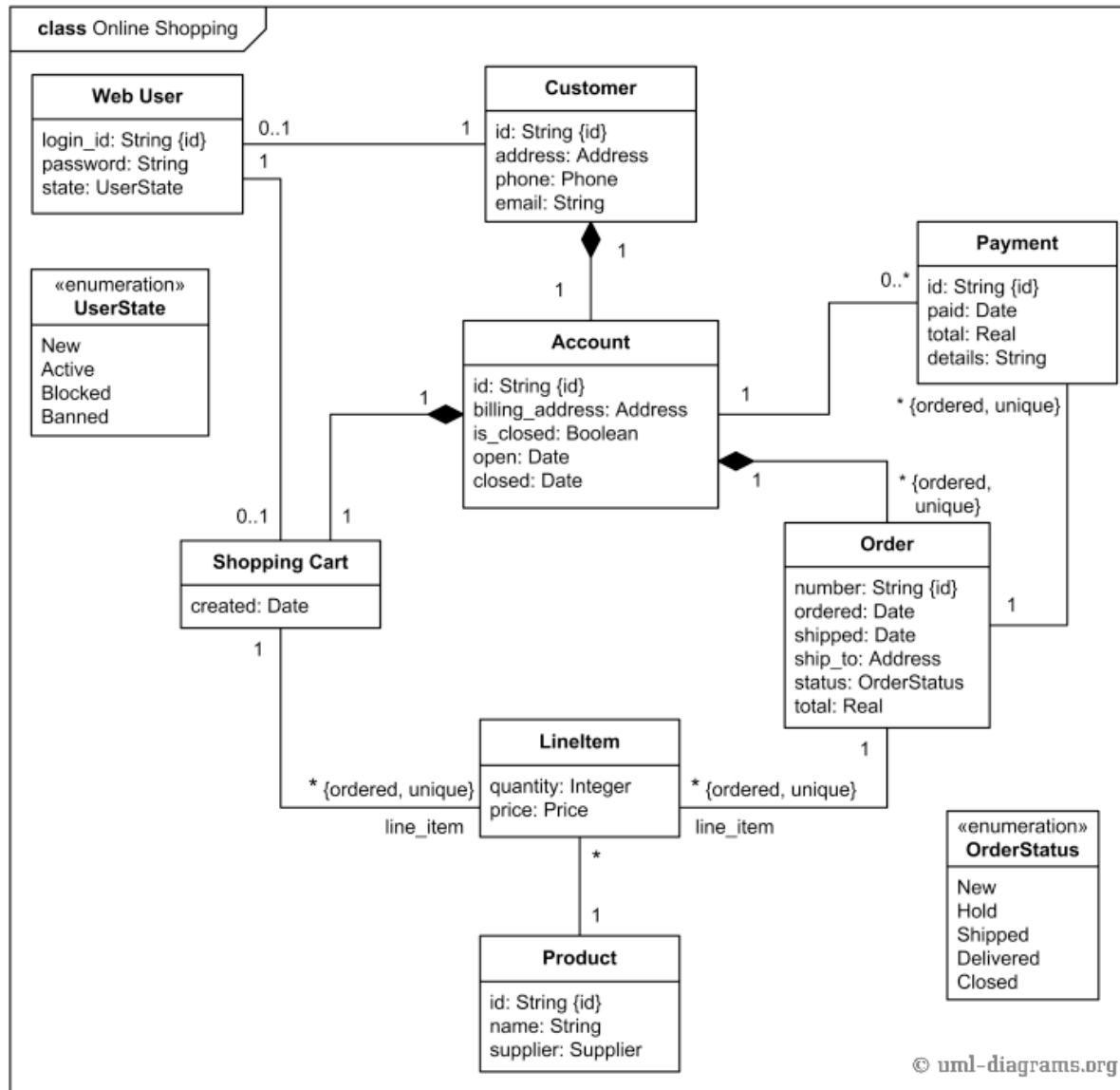


Diagramme de packages

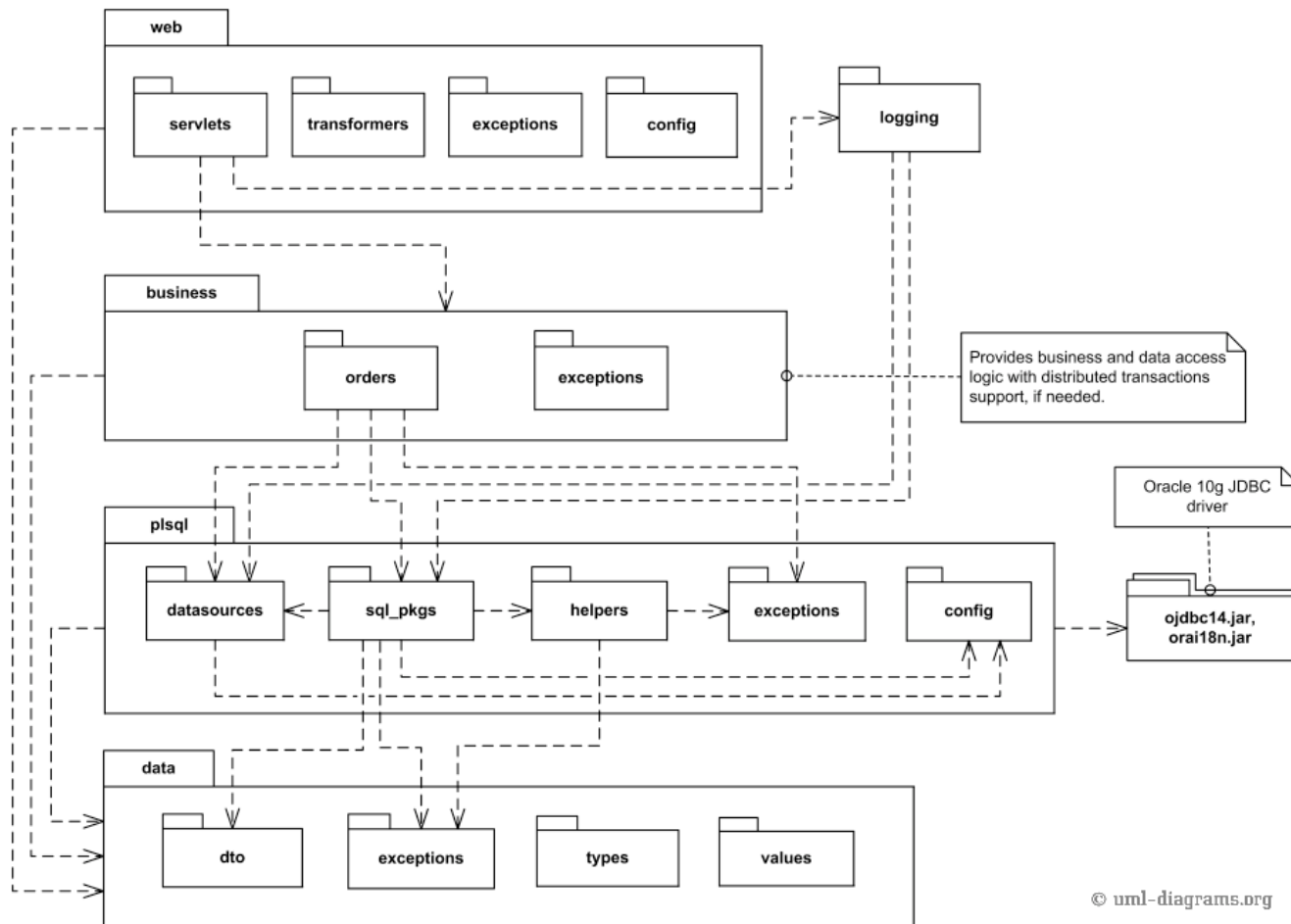


Diagramme de composants

(option ISA-DEVOPS au S8)

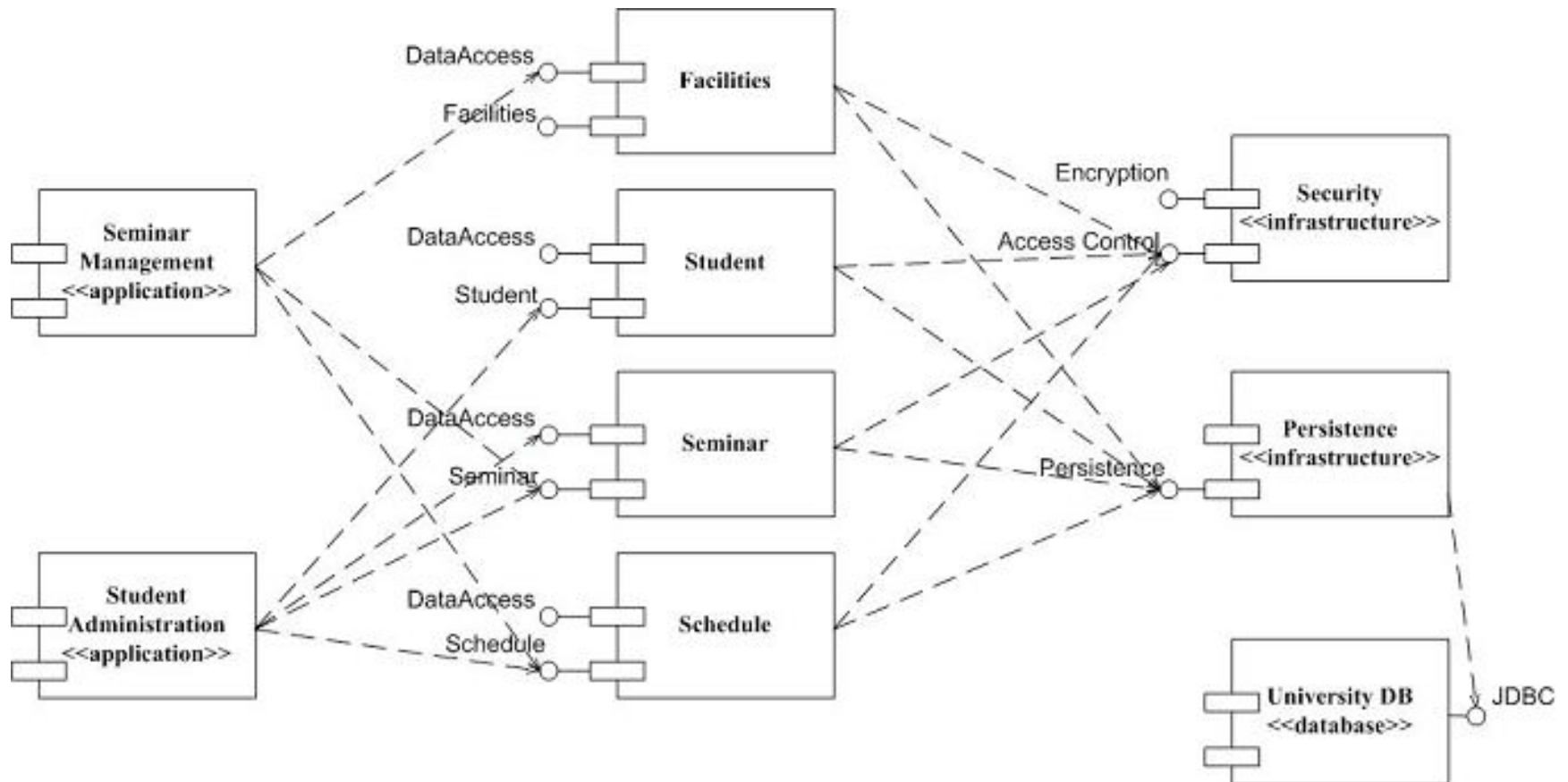


Diagramme de déploiement (option ISA-DEVOPS au S8)

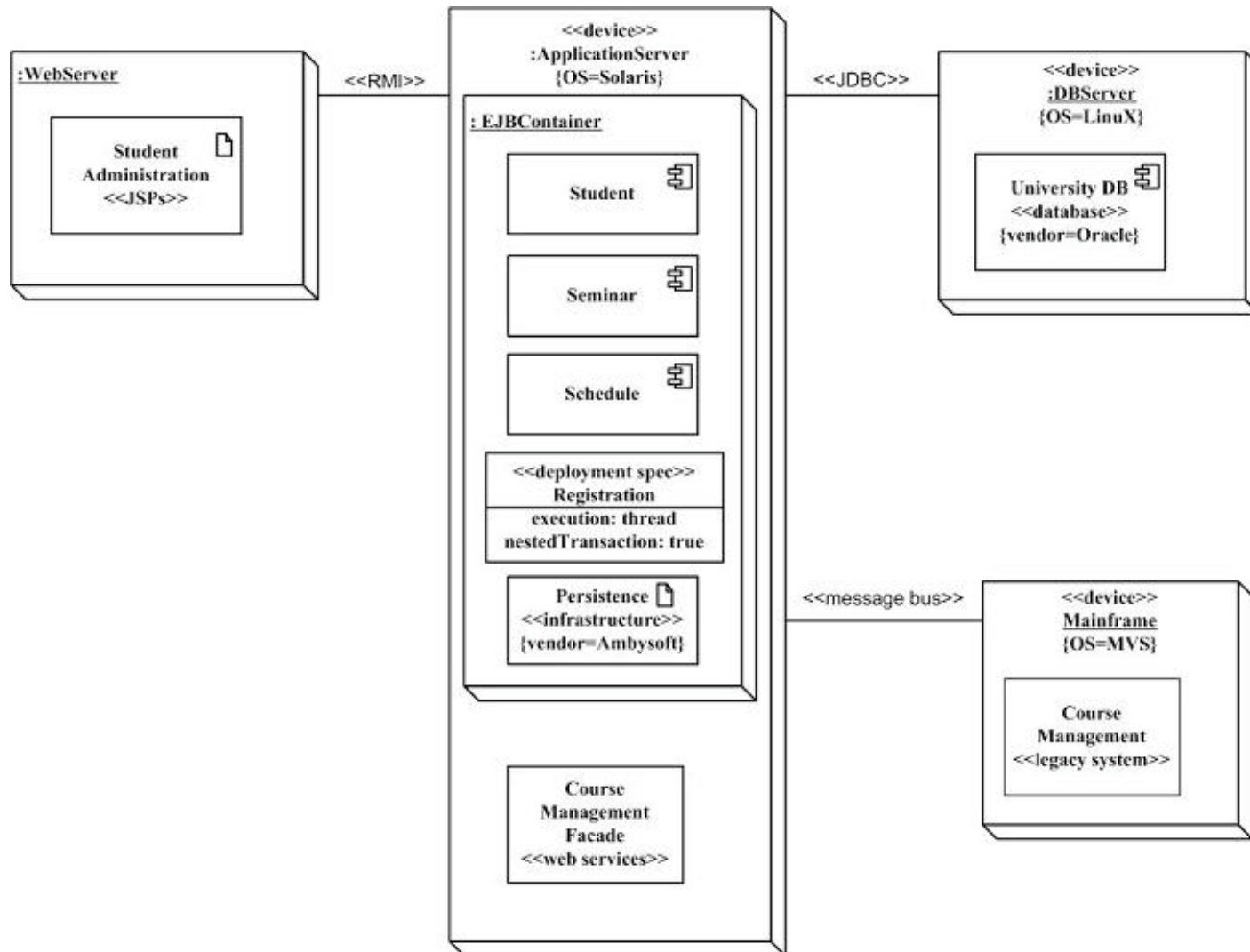


Diagramme d'états

(cf. cours Finite State Machines + slides support)

