Description des systèmes

Modélisation avec SysML & UML

Sommaire

1. Préambule
2. Introduction à SysML
3. Aspect comportemental
4. Aspect structurel
1. Exigences

1. Préambule
Activités proposées

Allers – retours entre :

- **Cours** :
  - Présentation du langage SysML
  - Les différents diagrammes et leur rôle
  - Les éléments des diagrammes

- **Travaux pratiques :**
  - Activités sur poste
  - Modélisation d'un système support

- **Débats :**
  - Echange des points de vue

1. Préambule
Exemple de système

- **ArDrone** :

Quadricoptère piloté par un iPhone/iPod-Touch/iPad
ainsi que par la plupart des appareils mobiles Wi-Fi basés sur Android

2. Introduction
L'ingénierie système – Qu'est-ce ?

- L'**Ingénierie Système** (**IS**), ou Systems Engineering en anglais (SE) : démarche méthodologique pour répondre à des problèmes complexes par la réalisation de solutions logicielles et matérielles
- Secteurs de l'activité industrielle concernés :
  - Systèmes embarqués :
    - Automobile
    - Ferroviaire
    - Aéronautique
    - Espace
    - Militaire
  - Télécoms
  - Santé/médical
  - Production d'énergie…

2. Introduction
L'ingénierie système – Le constat

- Les méthodes de l'Ingénierie Système (IS) :
  - **modélisation** pour valider les exigences
    - représentations concrètes avec des plans ou modèles réduits
    - plus abstraites avec des systèmes d'équations
  - **Simulation** pour vérifier ou évaluer le système.

- Aspects du système à modéliser :
  - décomposition fonctionnelle
  - flux de données
  - décomposition structurelle

- Spécifications issues de l'IS :
  - documentation dense due à une **approche orientée documentation** (« document-based approach »),
  - sélection inconsistante de différents types de diagrammes

2. Introduction
L'ingénierie système – L'évolution

- L'alternative : transition vers une « **approche orientée modèles** » (« model-based systems engineering » ou MBSE)

- Réalisation d'un *ensemble organisé* de modèles,
  - aspect opérationnel (contexte et utilisation du système),
  - aspect fonctionnel (structure et sous-fonctions du système)
  - aspect physique (architecture).

- La modélisation permet de maitriser la complexité du système étudié, car chaque modèle donne accès à une représentation abstraite de différents aspects du système.

2. Introduction
Les langages de modélisation

- Besoin émergeant >> Définition d'un système
- Concept abstrait >> Définition rigoureuse de produits
- Rôle de la modélisation >> Représentation pour :
  - Analyser
  - Concevoir
  - Réaliser ou simuler
  - Valider & justifier des choix
  - Communiquer

- **UML :** Langage de modélisation orienté **logiciel**
- **SysML :** Langage de modélisation orienté **systèmes** (pluritechnologiques)

2. Introduction

Un peu d'UML pour commencer

Présentation

Le modèle objet UML

La complexité du logiciel

La complexité des problèmes

- Les logiciels doivent parfois **traiter des éléments complexes** (par exemple, le pilote automatique d'un avion de ligne) auxquels viennent s'ajouter des exigences comme la facilité d'emploi, les performances …

La complexité du logiciel

La complexité des problèmes

- Ces contraintes forment un ensemble de **besoins** que l'utilisateur a **du mal à exprimer** à un concepteur de logiciels qui ne connaît pas nécessairement le domaine d'activité de l'utilisateur.
- Ce dialogue s'effectue aux travers de **documents** parfois volumineux et sujets à **interprétations**.

La complexité du logiciel

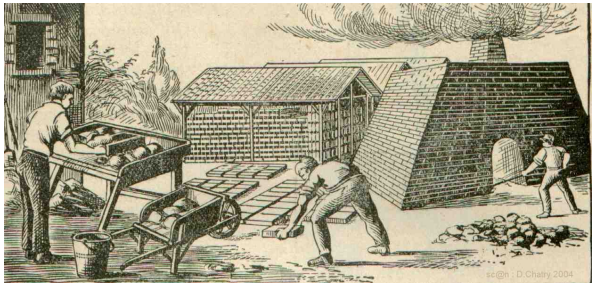
La difficulté à contrôler les processus de développement

- Les logiciels doivent parfois traiter (en interne) des **éléments complexes** tout en conservant (en externe) une relative **simplicité d'utilisation**. Cet objectif peut être atteint en utilisant, par exemple, des **interface utilisateurs** graphiques et intuitives.

La complexité du logiciel

La flexibilité dans la programmation

- L'industrie du logiciel possède moins de normes, les concepteurs sont alors tentés de créer leurs propres **briques de base** pour s'assurer qu'elles répondent parfaitement à leur **besoins** : le développement logiciel en devient donc d'autant **plus laborieux**.



L'arrivée d'UML

La normalisation

- **UML** devient une norme de l'OMG en **1997**.

- L'OMG (Object Management Group) est un organisme à but non lucratif créé en 1989 afin de promouvoir des standards qui garantissent la communication des applications orientées objet développées sur des réseaux hétérogènes.

- Cet organisme a été créé et est soutenu par des industriels comme HP, Sun, Unisys, American Airlines, Philips …

L'arrivée d'UML

Au final, qu'est-ce qu'UML ?

- **UML** est un **langage visuel**, basé sur l'utilisation de **diagrammes normalisés**.

- Il permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.

- UML **n'est pas une méthodologie** de développement, contrairement à RUP (*Rational Unified Process*).

L'arrivée d'UML

Les différents diagrammes

- **UML** (qui est avant tout un langage graphique) propose 13 types de diagrammes.

- Ces diagrammes sont présentés dans la norme sous forme d'un diagramme de classes afin de mettre en évidence les deux types de diagrammes :
  - les diagrammes de structure pour modéliser l'aspect **statique** d'un système ;
  - les diagrammes de comportement pour modéliser l'aspect plutôt **dynamique** d'un système.

L'arrivée d'UML

Les points forts

- UML est un langage formel et normalisé :
  - un gain de précision (pas d'ambigüité) ;
  - un gage de stabilité qui encourage à la création et l'utilisation d'outils.

- UML est un support de communication performant :
  - il cadre l'analyse ;
  - il facilite la compréhension de représentations abstraites complexes ;
  - son caractère polyvalent et sa souplesse en font un langage universel.

L'arrivée d'UML

Les points faibles

- L'utilisation pratique d'UML passe par un apprentissage.
- **UML ne propose pas de méthodologie, c'est seulement un langage graphique**.

2. Introduction

Un peu de SysML pour continuer

2. Introduction
Présentation de SysML

- **Historique**

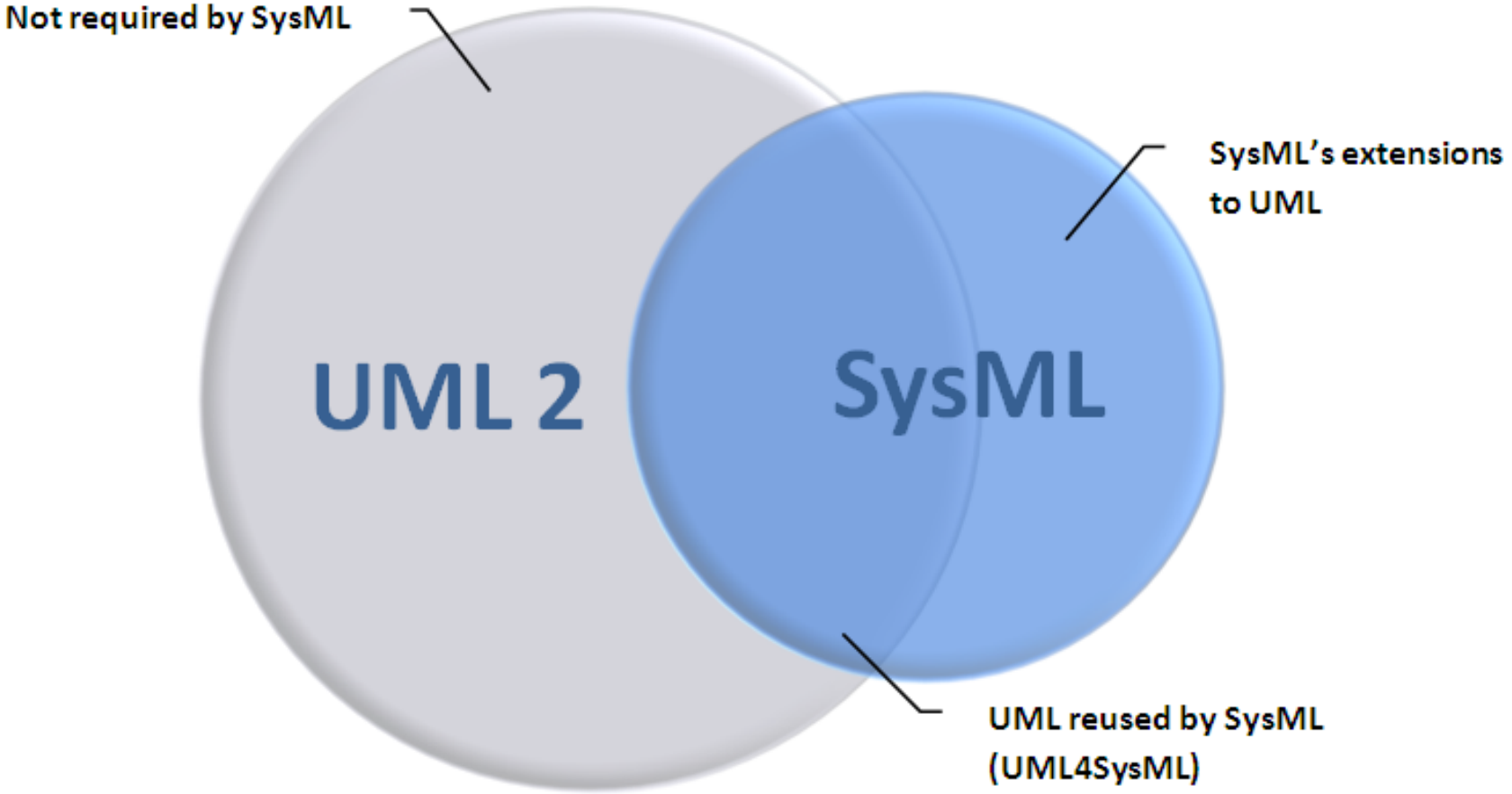
- Besoin de définir un langage basé sur UML pour l'IS initié en 2001 par l'organisation internationale de l'ingénierie système **INCOSE** (International Council on Systems Engineering)

- Plusieurs membres de l'industrie (BAE, Motorola, Boeing.), éditeurs d'outils (IBM, Sparx Systems.), universités et organisations ont travaillé sur la définition du langage.

- Chronologie :
  - Juillet 2006 : l'OMG annonce l'adoption de SysML
  - Septembre 2007 : spécifications de la version 1.0 rendues officielles
  - Décembre 2008 : SysML v1.1
  - Juin 2010 : SysML v1.2
  - Juin 2012 : SysML v1.3 (version actuelle)
  - Les spécifications de SysML, tout comme UML, sont disponibles gratuitement en anglais depuis le site de l'OMG : **www.omg.org** ou **www.sysml.org**

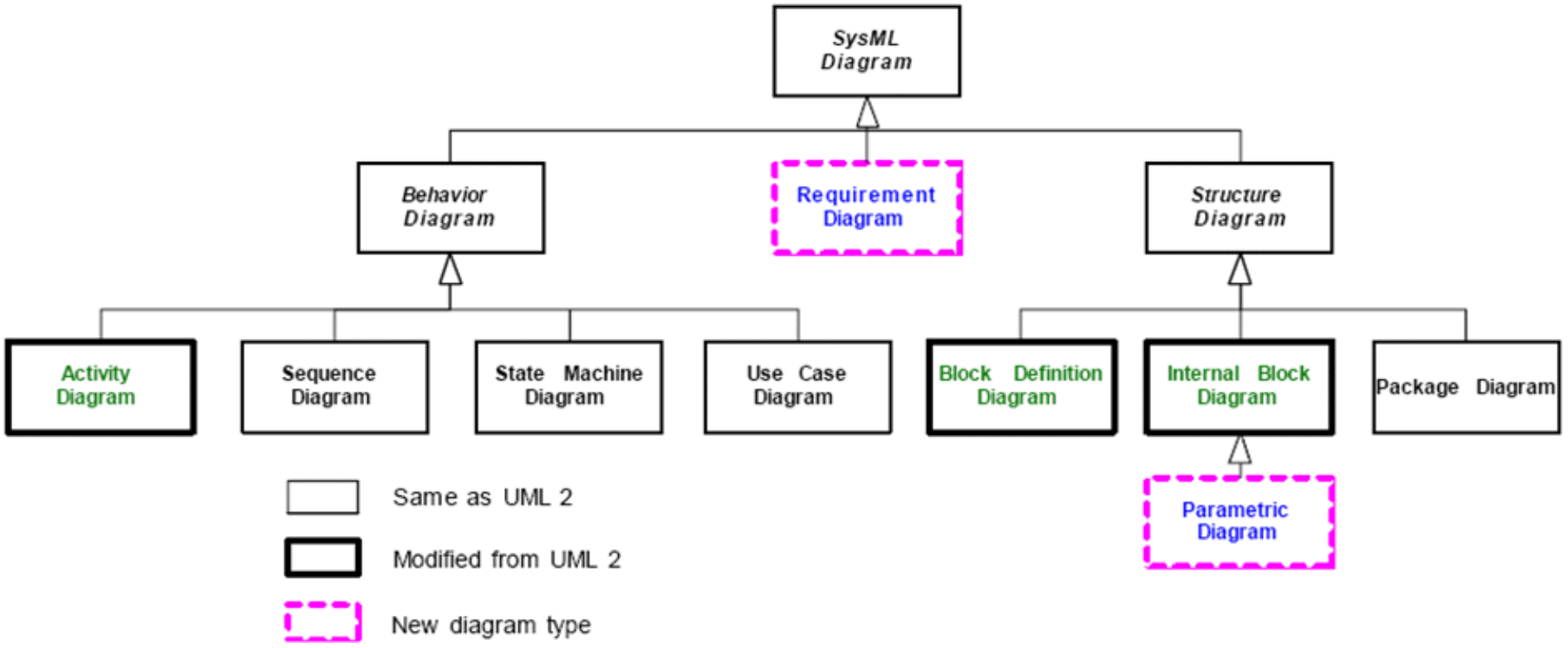
2. Introduction
Présentation de SysML

- **Rapport avec UML**



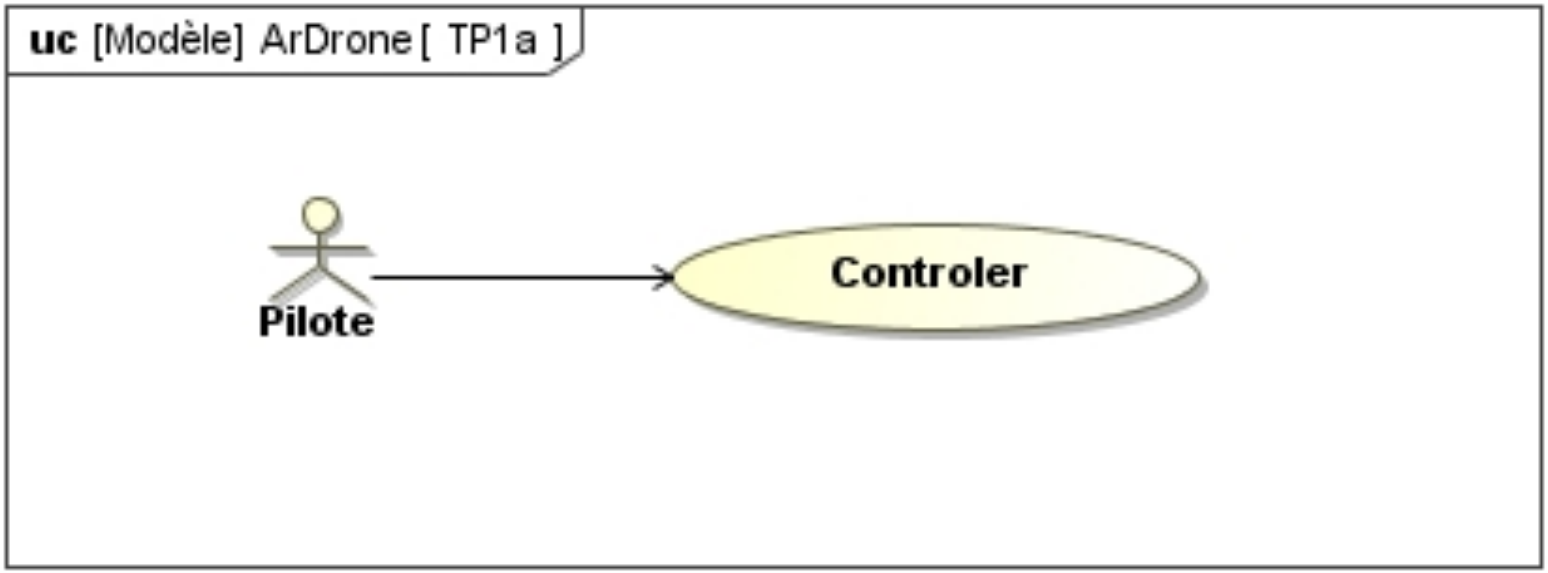
2. Introduction
Présentation de SysML

- **Classification des diagrammes**



2. Introduction
Présentation de SysML

- **Cartouche des diagrammes**



2. Introduction
Présentation de SysML

- **Ressources documentaires:**

- Ebook : SysML par l'exemple – Pascal ROQUES – Eyrolles
- SysML tutorial – www.incose.org
- Modélisation SysML – Guillaume FINANCE – uml.developpez.com
- www.sysmlforum.com
- www.afis.fr
- www.uml-sysml.org/sysml

2. Introduction
Présentation de SysML

- **Ressources logicielles**

Commercial SysML Tools	Open Source SysML Tools
• Enterprise Architect + MDG Technology for SysML (Sparx Systems) • MagicDraw + SysML plugin (No Magic) • UModel Enterprise Edition (Altova) • Rational Rhapsody Developer (IBM) • Artisan Studio (Atego)	• TOPCASED-SysML (TOPCASED Modeling Framework Open Source Project) • Papyrus for SysML (Papyrus Open Source Project) • Modelio Free Edition + Modelio SysML Designer module (Modelio-Open)