(Reference) Architecture = Components + Composition (+ Variation Points)?

Kung-Kiu Lau and Simone Di Cola

School of Computer Science
The University of Manchester
United Kingdom

kung-kiu@cs.man.ac.uk

Overview

- Different notions of architecture, component and composition in different communities
- We outline some fundamental characteristics of components and composition and posit their relevance to reference architecture

Architecture = Components + Composition

SA

- Architecture = many artefacts (including design)
- Components visible
- Composition visible?

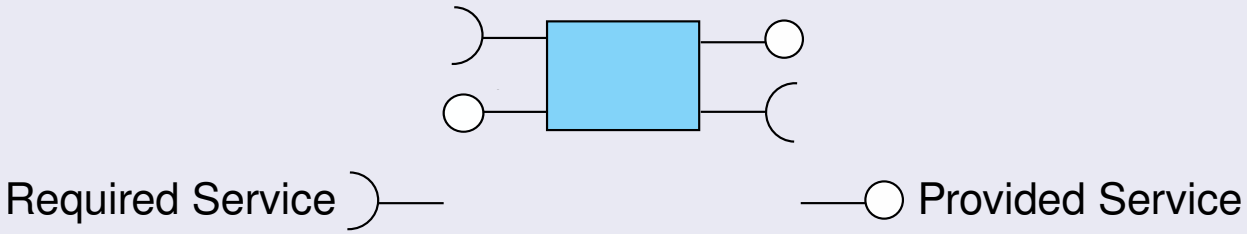
CBSE

- Architecture = structure of software system under construction
- Components = software units
- Composition = first-class citizen?
- Composition = glue?

Component Model ≡ Components + Composition

Component Models

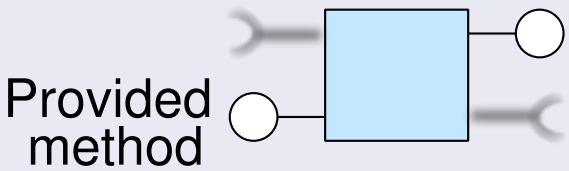
- Study of component models has highlighted the role of composition.
- Universally accepted view of a generic component:



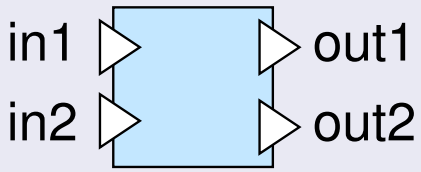
- No universal view of composition
- Composition still underdeveloped

Current Component Models

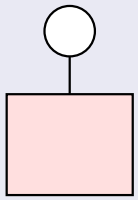
Components (first class entities)



(a) Object



(b) Architectural unit



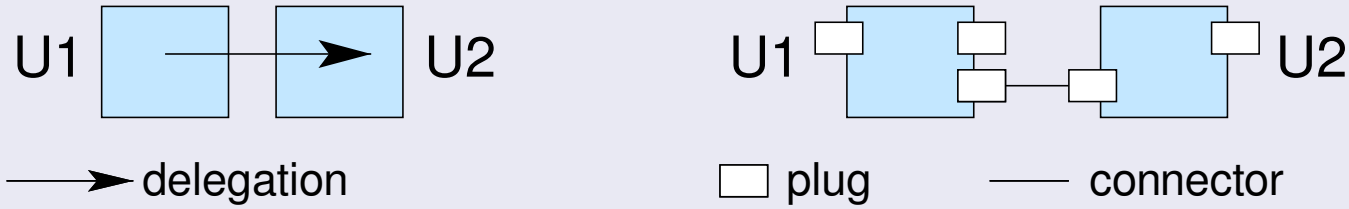
(c) Encapsulated component

Composition (not always first class entity)

Components	Provided services	Required services	Composition mechanism	Is composition first class?
Objects	Methods	——	Method call	✗
Architectural units	Out-ports	In-ports	Port connection	✗
Encapsulated components	Methods	*None*	Exogenous composition	✓

Composition

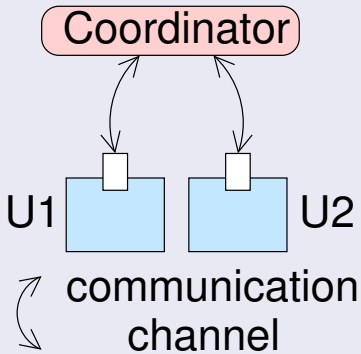
Composition by Connection: Method Call, Port Connection



(a) Direct message passing

(b) Indirect message passing

Composition by Coordination: Exogenous Composition



Architecture Description Language (ADL) = Component Model

ADLs: most widely used component models in CBSE

- Disadvantages:
  - composition not first-class
  - no explicit composition operators
  - composition defined at port level
- Advantages:
  - hierarchical (algebraic) construction
  - composite same type as constituent components
- Question: best candidate for reference architectures?

Variation Points = ?

How to define variation points in component models?

Components	Feature	Variation Point	Architecture
Objects	Object/Aspect	Method call/Pointcut	Class diagram?
Architectural units	Architectural unit	Port connection	Architecture template
Encapsulated components	Component	Variation operator	Master architecture

Issues

- How to map between features and objects/aspects/architectural units/encapsulated components?
- Are method call (and pointcut) and port connection good variation points?
- What kind of architecture (with what kind of variation points) is suitable as reference architecture?

Reference Architecture = Components + Composition + Variation Points?

Assumption

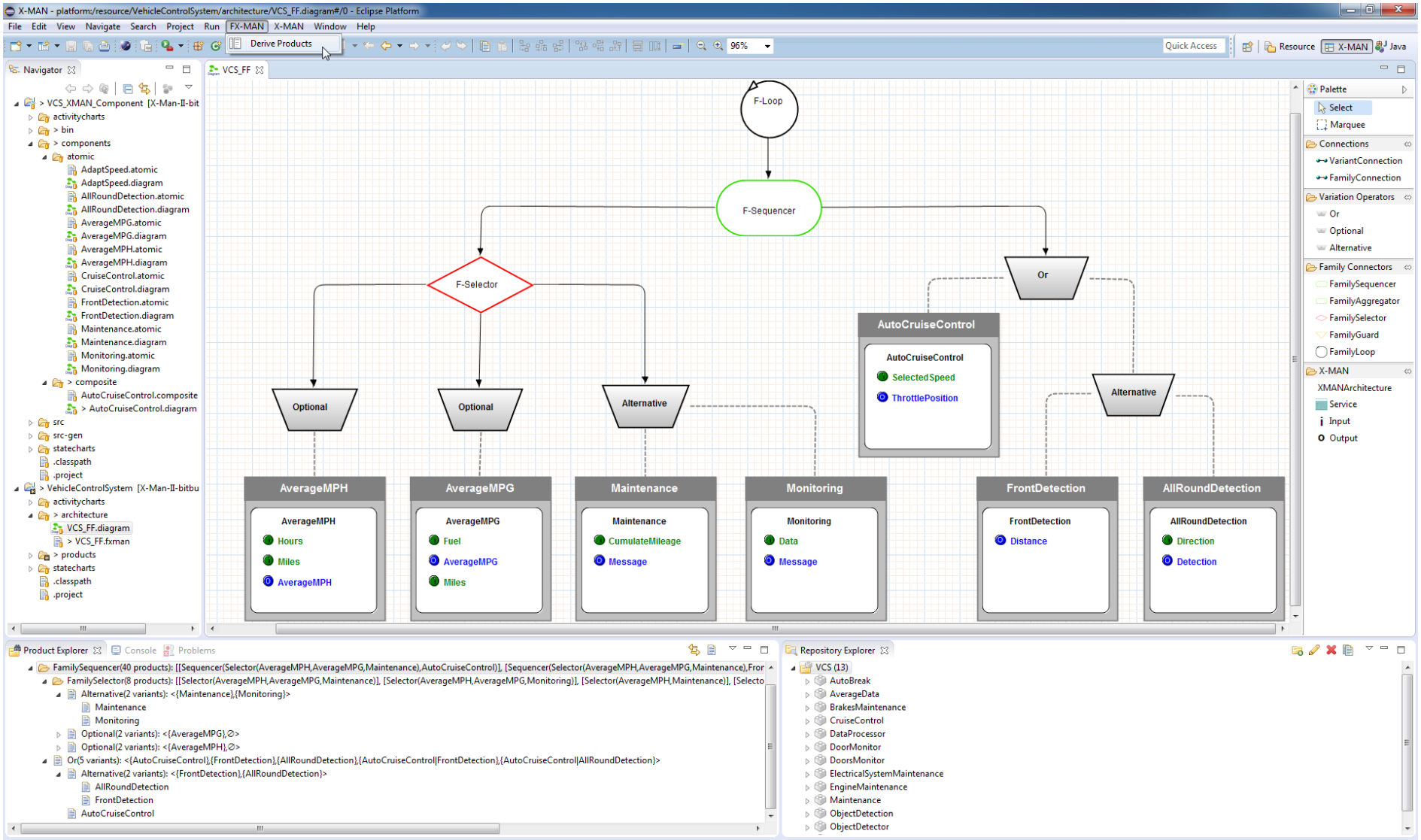
Reference architecture = architecture template for all possible products (in a domain) = master architecture

Conjecture

Variation points should be first-class entities in reference architecture that is a master architecture

Is this a Reference Architecture?

For Vehicle Control Systems



Feature Model for Vehicle Control Systems