WS72 Systems Architecture

Structured Methods II: Structured Design and Semantic Transformations

Lecture 11

Loughborough University accepts no third party liability for the contents of this lecture and gives no endorsement to any products, processes and services mentioned within.

Overview

Key Concepts

Modular design

Cohesion and coupling

Functional allocation

Synthesis

2nd order transformations

Key Topics

- Structured Design
- Structural Type
- Model Specification and Semantic Transformation
- Essential System Architecture

Structured Design

- Principle of Structured Design
  - Systems should be comprised of modules …
    that each are highly cohesive
    but collectively are loosely coupled
- Cohesion minimises functional relationship between elements not in the same module
- Coupling is minimised between modules
- Also, the solution should reflect the inherent structure of the problem*.

Just Enough Structured Analysis by the late Ed Yourdon was available from http://www.yourdon.com/jesa.
A PDF can be found at http://zimmer.csufresno.edu/~sasanr/Teaching-Material/SAD/JESA.pdf (accessed 19th September 2018)
Since his death in 2016, Wikipedia has become a useful place to find his works: https://en.wikipedia.org/wiki/Edward_Yourdon

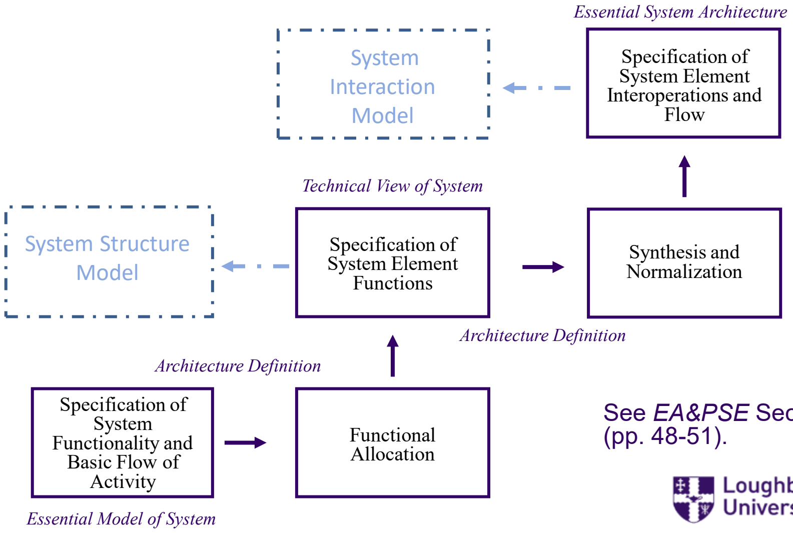
Note that this is accomplished by semantic transformation.

Levels of Cohesion Distinguished by Types of Association*

▪ Coincidental	*concurrence without apparent cause*
▪ Logical	*e.g. association by conjunction (p and q)*
▪ Sequential	*association by order (e.g.1 precedes 2)*
▪ Temporal	*association by time order*
▪ Communication	*exchange of information*
▪ Procedural	*related by a specified 'way of doing'*
▪ Functional	*bound by a 'way of doing' for purpose*
▪ Aggregation	*formation of multiple constituents into one*
▪ Composition	*aggregation of constituents to form a whole*

These levels elaborate those of Yourdon. The descriptions are not formal but rather are suggestive of an increasing formation of a unified whole.

A Framework for Structured Design



Essential System Architecture

System Interaction Model

Specification of System Element Interoperations and Flow

Technical View of System

System Structure Model

Specification of System Element Functions

Synthesis and Normalization

Architecture Definition

Architecture Definition

Specification of System Functionality and Basic Flow of Activity

Functional Allocation

Essential Model of System

See *EA&PSE* Section 4.3 (pp. 48-51).

Loughborough University

Overview

Key Concepts

Modular design

Cohesion and coupling

Functional allocation

Synthesis

2nd order transformations

Key Topics

- Structured Design
- ***Structural Type***
- Model Specification and Semantic Transformation
- Essential System Architecture

Loughborough University

Model Specification and Transformation*

- Engineering is concerned with *Concept Realisation***
  - The concepts are abstractions that refer to a system of interest
  - Interpretation of concepts into models is part of realisation
  - Structured Methods I gave a ‘concrete’ method for a process
  - Understanding the foundation will help you to put it into practice
- An *Abstract Architecture* approach is required
  - Architecture is at a higher level of abstraction than engineering
  - A structured viewpoint is taken on the technical processes

→ The outcome is a specification of graphical models and transformations between them for structured concept realisation.

*Methods were first published in 2013 [Link 16 paper]; refined in 2020 [IEEE Architecture Definition paper].

**Refer to the definition of engineering in the Terms of Reference lecture.

Architecture as Structural Type [1]

The mathematical foundation of Architecture Definition

- Architecture Definition requires a precise but practical definition of the term *architecture* for use in engineering [1]
  - The ISO/IEC/IEEE definition adopted in 2011 has limitations [2]
  - ISO 2011 concept of *embodiment* can be confused with *model*
  - ISO/IEC/IEEE 42020, and 42030:2019 consider refinements
- The essential definition considers *architecture* as [1]
  - Structural type in conjunction with compatible properties, i.e. consistent with the type; implementable in a class of the type
  - Architecture can be regarded as a coupled pair (type, properties)
  - Similar to (set, structure) used in mathematics for ‘spaces’

[1] Dickerson, et al 2020 (IEEE Architecture Definition). See Chpt 3 references.

[2] ISO/IEC/IEEE 42010:2011, and used in ISO/IEC/IEEE 15288:2015

Precise Language for Structural Types

- Blending the language of set theory and software engineering:
  - Mathematical objects are abstractions that possess properties [1]
  - Classes of objects *are defined by* shared properties (set theory)
  - Equivalently, properties *are implemented by* classes (e.g. UML) [2]
  - Classes *are realised by* sets (i.e., are given logical existence)
  - Type is a specialised property (e.g., a monadic predicate) [1]
  - →Architecture is *implemented* by classes and *realised* in structures
- *An Architecture is* [1, 3]

  A class of structure in which (architectural) properties can be implemented. The objects of an architecture class are structures into which *interpretations* of concepts are realised (e.g., models).

[1] Dickerson, et al 2020 [IEEE Architecture Definition paper]

[2] In other words, classes *implement* properties.

[3] A structure is a set. A collection of structures of a *specified type* is a class. Properties that can be implemented in the class are called *architectural. Interpretations* realised in the structures are 'blueprints' for implementation.

Types and Properties for an Essential Model

- An *Essential Model* can be specified using 4 structural types
  - *Partition* (decomposition of a set into pairwise disjoint subsets)
  - *Association* (e.g. binary pairing, ordered pairs)
  - *System Hierarchy* (representation of structure using partitions) [1]
  - *System Flow* (a transitive ordering and logical control structure)
- Architectural properties implemented in the 4 structural classes: [2, 3]
  - *Separation* (e.g. assign system, environment to disjoint sets)
  - *Functionality* expressed through *interaction* (an association)
  - *Behaviour* expressed through *activity state* (idle or active)
- Architectural concepts can be realised in UML graphical structures

[1] INCOSE SEH 4th ed,, section 2.3 p.7

[2] Refer to Dickerson, et al 2020 (IEEE reference)

[3] More than one class of structure may be needed to implement a class of architecture, e.g. functional architecture is implemented by partitions and associations.

Overview

Key Concepts

Modular design

Cohesion and coupling

Functional allocation

Synthesis

2nd order transformations

Key Topics

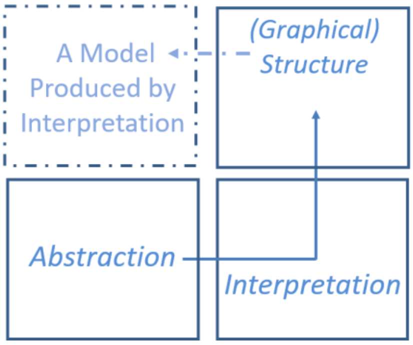
- Structured Design
- Structural Type
- ***Model Specification and Semantic Transformation***
- Essential System Architecture

Loughborough University

Semantic Transformation:
*Recall the Proposed Definition**

- Transformation of data to add semantic knowledge
- Klir Methodology

  Use interpretation to increase information content
- *Traceability between models*
  - *Every word and its meaning*
  - *Every relationship*



Semantic transformation *is a technique for interpretation of a model of (or related to) a system into a semantically richer model by using a specified set of modeling and structuring rules.*

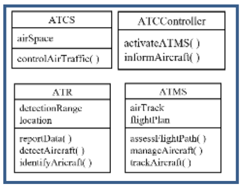
* Ref [IEEE SoSE, Norway 2016]

Loughborough University

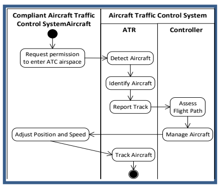
System Structure Model (1 of 2)
System Elements (White Box)

System Set (Elements)

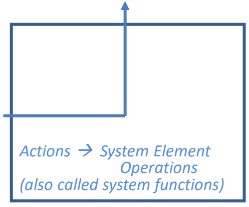
A system is a set of interrelated elements that comprise a whole, together with an environment.



Scope of Concerns: System Elements and their Operations


Compliant Aircraft Traffic Control SystemAircraft
Aircraft Traffic Control System
ATR
Controller
Request permission to enter ATC airspace
Detect Aircraft
Identify Aircraft
Report Track
Assess Flight Path
Adjust Position and Speed
Manage Aircraft
Track Aircraft

Activity Model



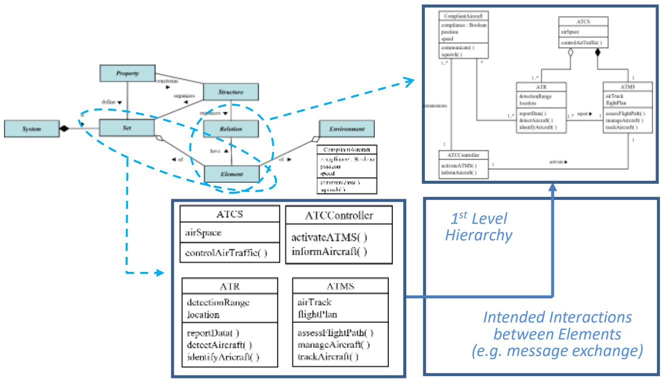
Actions → System Element Operations (also called system functions)

Transformation Rules: Functional Allocation

Loughborough University

System Structure Model (2 of 2)
Interactions of the System Elements

Class Diagram



1st Level Hierarchy

Intended Interactions between Elements (e.g. message exchange)

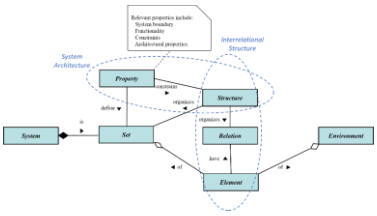
System Set (Elements)

Scope of Concerns: System Elements and Interactions

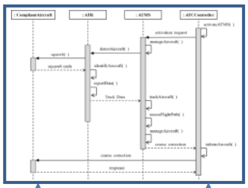
Transformation Rules: System Hierarchy Message Exchange

Loughborough University

The Essential System Architecture (Implemented in an Interrelational Structure)

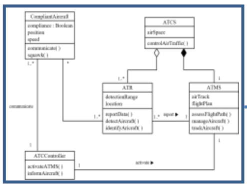


Essential Sequence Diagram



Scope of Concerns: Order of Functional Flow and Message Exchange, Consistency

Transformation Rules: Synthesis of Sequencing with Activity Flow and Message Exchange

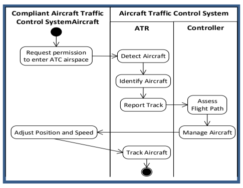


Class Diagram

Activity Flow → Sequencing

Class → Object → Life Line

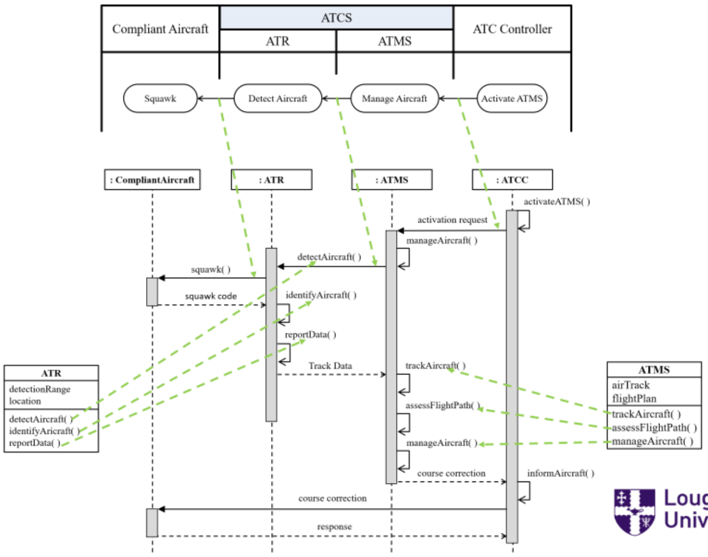
Operations → Function Calls and Exchanges


Compliant Aircraft Traffic Control SystemAircraft
Aircraft Traffic Control System
ATR
Controller
Request permission to enter ATC airspace
Detect Aircraft
Identify Aircraft
Report Track
Assess Flight Path
Adjust Position and Speed
Manage Aircraft
Track Aircraft

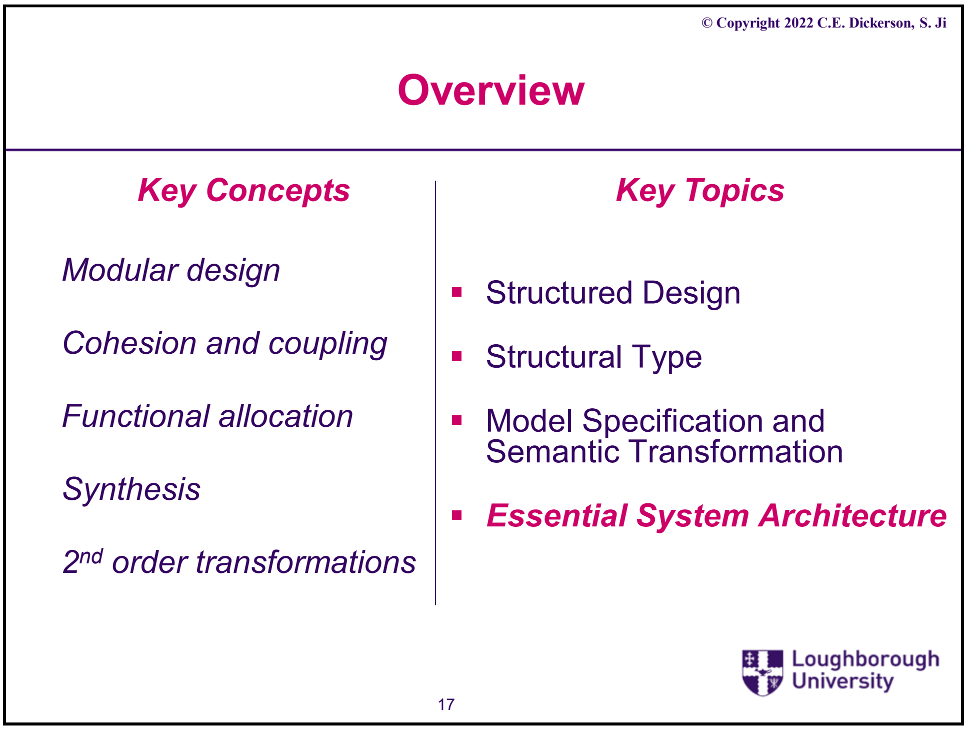
Activity Diagram

Loughborough University

Example of Synthesis from Tutorial II



Loughborough University



Overview

Key Concepts

Modular design

Cohesion and coupling

Functional allocation

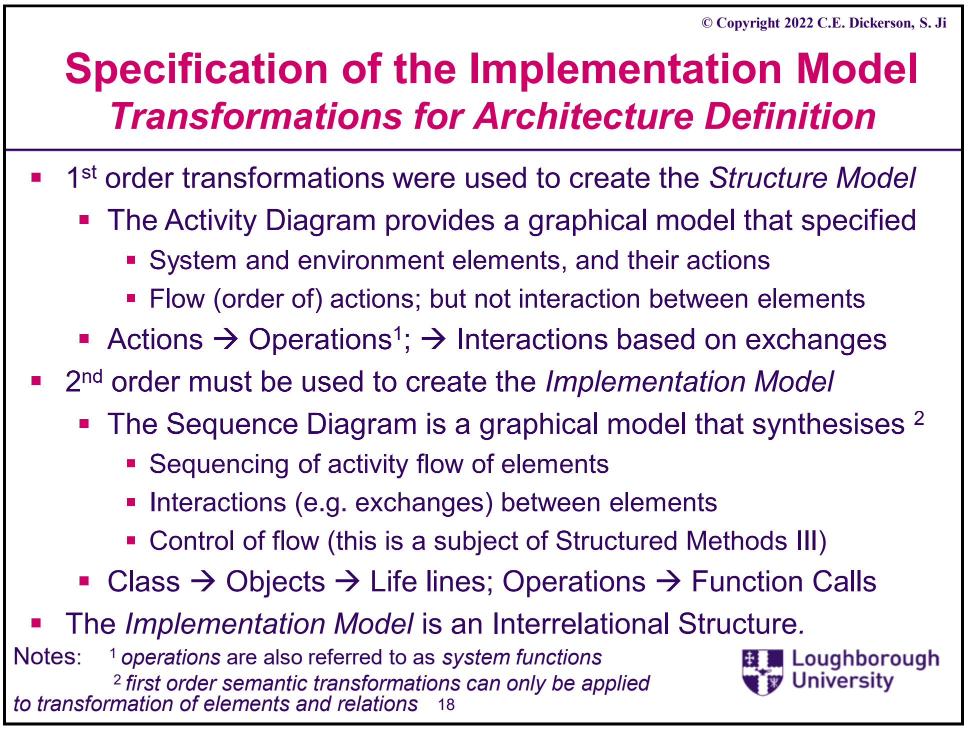
Synthesis

2nd order transformations

Key Topics

- Structured Design
- Structural Type
- Model Specification and Semantic Transformation
- ***Essential System Architecture***

Loughborough University



Specification of the Implementation Model
Transformations for Architecture Definition

- 1st order transformations were used to create the *Structure Model*
  - The Activity Diagram provides a graphical model that specified
    - System and environment elements, and their actions
    - Flow (order of) actions; but not interaction between elements
  - Actions → Operations[1]; → Interactions based on exchanges
- 2nd order must be used to create the *Implementation Model*
  - The Sequence Diagram is a graphical model that synthesises [2]
    - Sequencing of activity flow of elements
    - Interactions (e.g. exchanges) between elements
    - Control of flow (this is a subject of Structured Methods III)
  - Class → Objects → Life lines; Operations → Function Calls
- The *Implementation Model* is an Interrelational Structure.

Notes: [1] *operations* are also referred to as *system functions*
[2] *first order semantic transformations can only be applied to transformation of elements and relations*

Loughborough University

Summary of Processes and Methods

Essential System Definition is concerned with how elements associated with a system can be defined in terms of purpose (functionality), behaviour, and inclusion. These can be specified in terms of properties of and relations between the elements.

Essential Architecture Definition is concerned with defining the structures associated with a system and its properties; relations amongst the structures, synthesis and normalisation; and interfaces between the system elements.

Architecture Implementation is concerned with (i) how the structures associated with a system respond under intended and alternative conditions, (ii) how control structures can be used for precise implementation of responses.

The output of the ETP processes is an architecture that is robust and enables system design: a software architecture that can be deployed to software developers; hardware element and interface definitions for system developers; and an implementation model for deployment to state machines (which will define how structures associated with the system change state in response to the occurrences of events).

ETP: Essential Technical Process

Loughborough University

Questions?

Loughborough University