Introduction to Enterprise Architecture

Associate Professor Peter Bernus

Griffith University

Course overview

1 Introduction To Enterprise Architecture

2 Modelling Frameworks, Enterprise Models And Enterprise Modelling Languages

3 Decisional Modelling and Reference Models

4 The Virtual Enterprises Challenge

5 Strategy and Performance Management

6 Preliminary (Architectural) Design (Master Planning Principles)

7 Enterprise-Modelling as an Act of Communication

8 Ontologies and the Selection of Enterprise Modelling Languages

Overview of today's lecture

- Enterprise Integration / Enterprise Architecture
- Architecture Frameworks – history and state of the art
- GERAM / ISO IS 15704:2000 & 2006
  - life-cycle and life history dimensions
  - enterprise modelling / views
  - relationship between enterprise entities
  - enterprise networks and virtual enterprises
- Research and development directions

Enterprise Integration / Enterprise Architecture (EI/EA)

Some problems that EI/EA as a discipline wants to address:

- Integration of information and material flow
- Management of change / evolution / relationships
- Management of human / technological / economic / environmental issues
- Gap between strategy and implementation

Enterprises are complex systems, and we must be able to account for some of their important characteristics:

- Change is *Dynamic*
- Change may be *Organic*
- The enterprise is a *Socio-technical* system, and nowadays the *ecology* (natural environment) can not be ignored either

- The initial tenet of EI/EA was that complex enterprises can be ‘designed & developed’ using suitably selected methodologies and tools (we called this *‘enterprise engineering’)*

- In this sense enterprise integration / Enterprise architecture was initially considered to be a special case of systems engineering, where the enterprise is the system being ‘designed & developed’

- However, this development does not happen purely through a sequence of deliberate design and implementation steps that we are used to in case of small- or moderate scale technical systems

  Enterprises develop through a combination of
  - *deliberate* action (design & development), and
  - (some) form of *self-evolution*

Therefore we need to…

1. Understand the nature of the problem, by *organising our knowledge* about enterprises and how they evolve

2. Create a *framework* that can be used to understand and to manage the evolution of enterprises

3. Demonstrate the use of the framework and how it can help various stakeholders who are concerned with / effected by the evolution of enterprises

4. Identify knowledge gaps to help the discipline evolve

Concept of Enterprise Architecture - Who' Doing It?

- EA is increasingly used by large, complex organizations in all sectors to integrate strategic, business, and technology planning.
  - Public Sector: Mandated in law for U.S. federal agencies -- several frameworks exist. States require agencies to have IT architectures. Limited use at the local level as best practice. South Korea now doing EA. Canada has adopted of the use of EA at the provincial level.
  - Private Sector: EA is used by leading companies in the U.S., but not shared as it is identified as an element that provides strategic competitive advantage. Most Fortune 500 companies have some form of EA in use….many use Zachman as a basis.
  - Non-Profit Sector: EA is a done as a best practice, and is a fairly new concept. Competition for donors and tight budgets will drive EA use in order to maximize IT resource effectiveness.
  - Academic Sector: EA is a done as a best practice. An increasing number of universities are now developing enterprise-wide architectures.
  - Military Sector: EA is required in the U.S. Department of Defense by DOD Directive 5000. All IT-related programs must use the "DOD Architecture Framework" (DODAF) for design and documentation, as well as DOD's extended version of the Federal EA Reference Models to report the status of major and mission-critical IT programs each year.

Concept of Enterprise Architecture – Why do it?

EA³

"It is my opinion that this issue of Enterprise Architecture is not well understood in the ranks of general management who see Enterprise Architecture as just an I/S or IT issue, nor in the ranks of I/S management who see it as taking too long and costing too much, nor in the ranks of academia who tend to focus on what they perceive constitutes current market demand, typically a promising technology. My opinion is, Enterprise Architecture may well be the "Issue of the Century." … I know I have a rather radical view of this, but my observation would be the whole reason you want people with technical skills in your Enterprise is not for building and running systems. Anybody can build and run systems, the employment of the technology. The reason you want these kinds of people in your Enterprise is because they have the capability of engineering and manufacturing your Enterprise for you. That's the reason for their being, NOT simply for building and running systems."

John Zachman, 2004

From Zachman's Foreword to "*An Introduction to Enterprise Architecture: 2nd Edition*" by Scott A. Bernard. AuthorHouse Publications, 2005. ISBN 1-4208-8050.

Architecture - Defined

EA³

Architecture: 1. the science, art, or profession of designing and constructing buildings, bridges, etc; 2. a building or buildings collectively; 3. a style of construction; 4. design and construction; **5. any framework or system**.

Webster's New World Dictionary. Simon and Shuster; 1980.

Organizations are *a type of system*:
(e.g., Businesses, Agencies, Military Units, Hospitals, Universities, Non-Profits)

Structures

Physical Things

Where People Work

EA uses a framework to document the architecture of large, complex human organizations. These organizations are goal-oriented social systems that are referred to as "enterprises"

Brick & Morter ► Click & Morter ► Click

Enterprises

Virtual Things

How People Work

'Architectures' developed in the past

- AFS that originated from manufacturing industry (Flexible mfg, CAD/CAM, CIM, …)
  - *GRAI* (U Bordeaux) – for integrating production mgmt [80s-…]
  - *CIMOSA* (Consortium/Association) – model based control in manufacturing [80s-…]
  - *PERA* (Consortium) – factory design / continuous process industries [90s]
- Others originated from IS / IT
  - *Zachman* – Information system design [end of 80s]
  - *TOGAF* – technical IT architecture US DoD [90s] → OpenGroup
  - *FEAF* – Federal Enterprise Architecture FRamework [USA, 1999]
- Others that originated from IS / IT
  - *C4ISR* –and derivatives (DoDAF, MODAF, NATO AF,… [00s]) military systems [90s]
  - *EA*^{3} – US Federal Gov
- etc...

All of these architecture frameworks consider the life cycle of the enterprise
In the early 1990s nobody really understood what architecture was…(!)

By Dennis E. Wisnosky - Engineering Enterprise Architecture: Call to Action, Public Domain, https://commons.wikimedia.org/w/index.php?curid=27153767

Other related efforts

- Systems engineering – definition of system engineering processes / ISO 15288 [since 90s], ISO42000 series [00s] (42010 – architecture descriptions, 42020 architecture processes, 42030 architecture evaluation)

- Software Engineering – definition of Software Engineering processes / ISO 12207 [since 90s]

- Not all architectures (AFs) proposed in the literature are ‘complete’ thus in practice we must mix and match their capabilities

- GERAM helps to identify how this can be done

Overview of past architecture efforts

- These architectures are in use today, or are the origins of present AFs
- Their contributions can be generalised
- Through this generalisation their developers can assess what they *do* and what the *do not* provide at the moment and can further develop these architectures to become more complete
- Practicioners can understand what architectures are for, what they need to contain, and through that be able to find their way in the complex word of architecture frameworks
- In the following, we shall review a few notable architecture frameworks

PERA

The Purdue Enterprise Reference Architecture

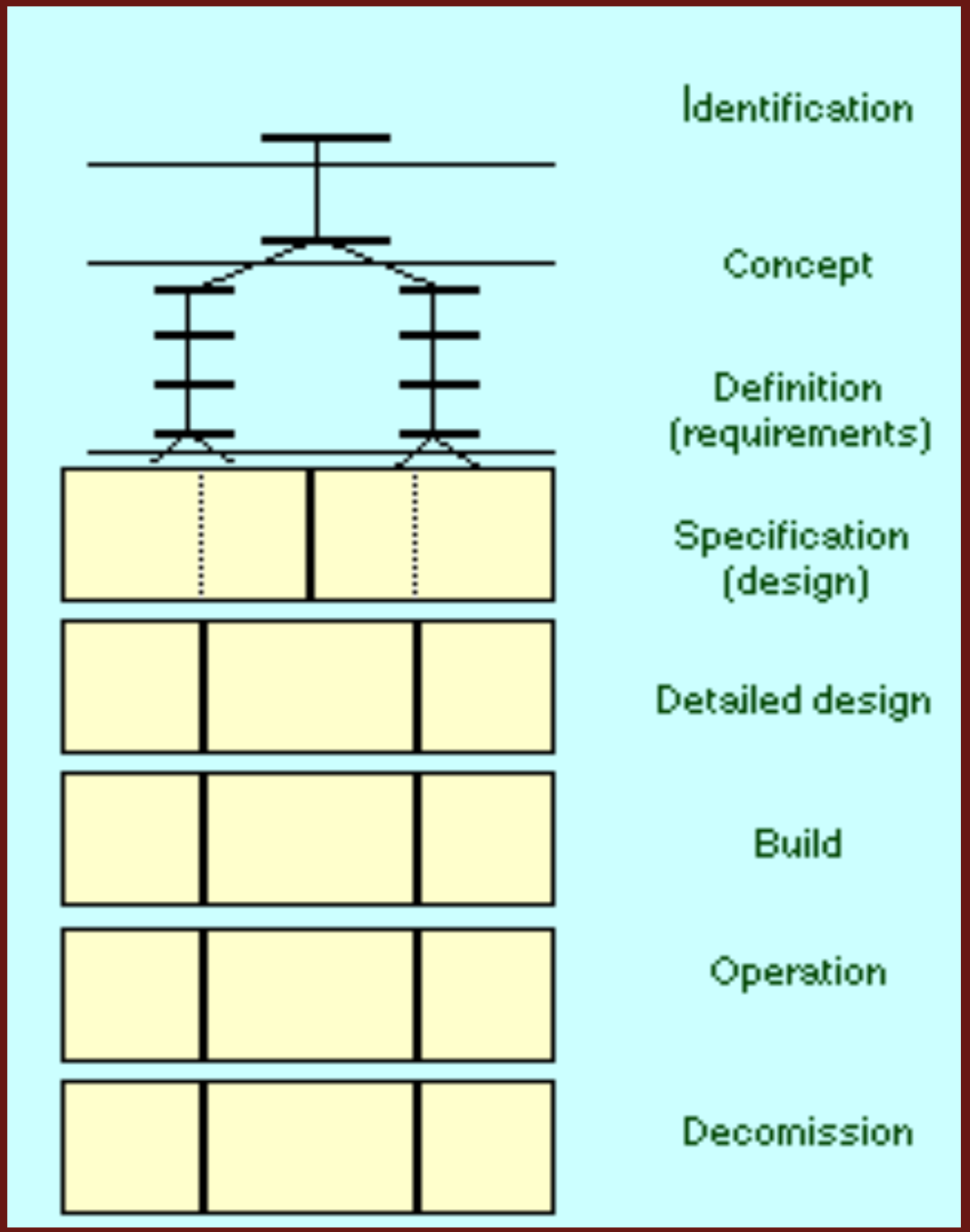
Developed by the Purdue Consortium (led by the Laboratory for Applied Industrial Control, Purdue University, Prof Theodore J . Williams)

PERA is of important historical relevance, as many concepts were first clarified in PERA

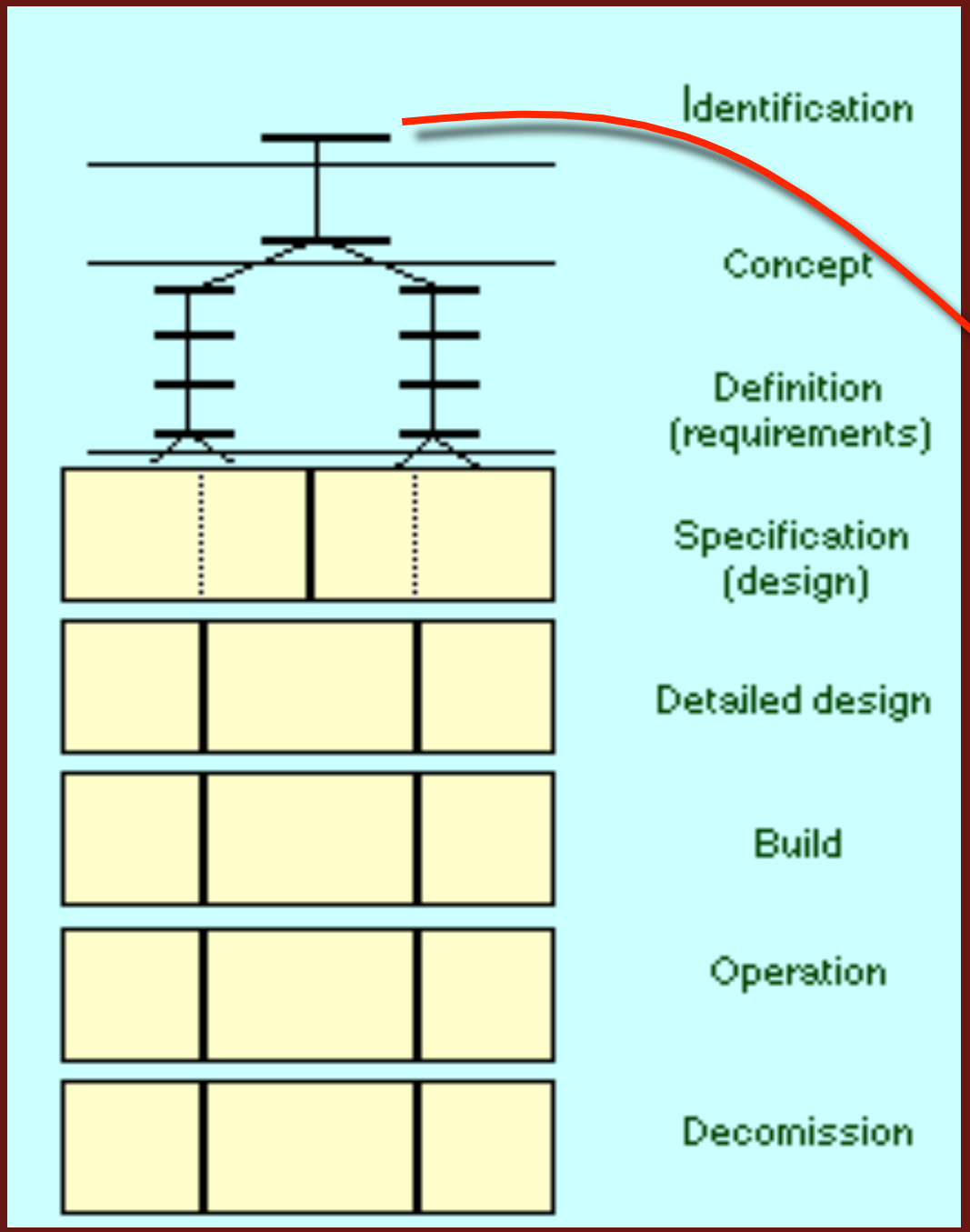
PERA is a life-cycle architecture

- Its scope is the entire enterprise, including all aspects and components and all activities from the beginning to the end of life

- See *Computers in Industry,* Vol. 24, No. 2-3, pp. 141-158 (1994) [available in the course Reading material]



The PERA life-cycle diagram organises the types of activities that need to be done during the life of an enterprise business entity (EBE)



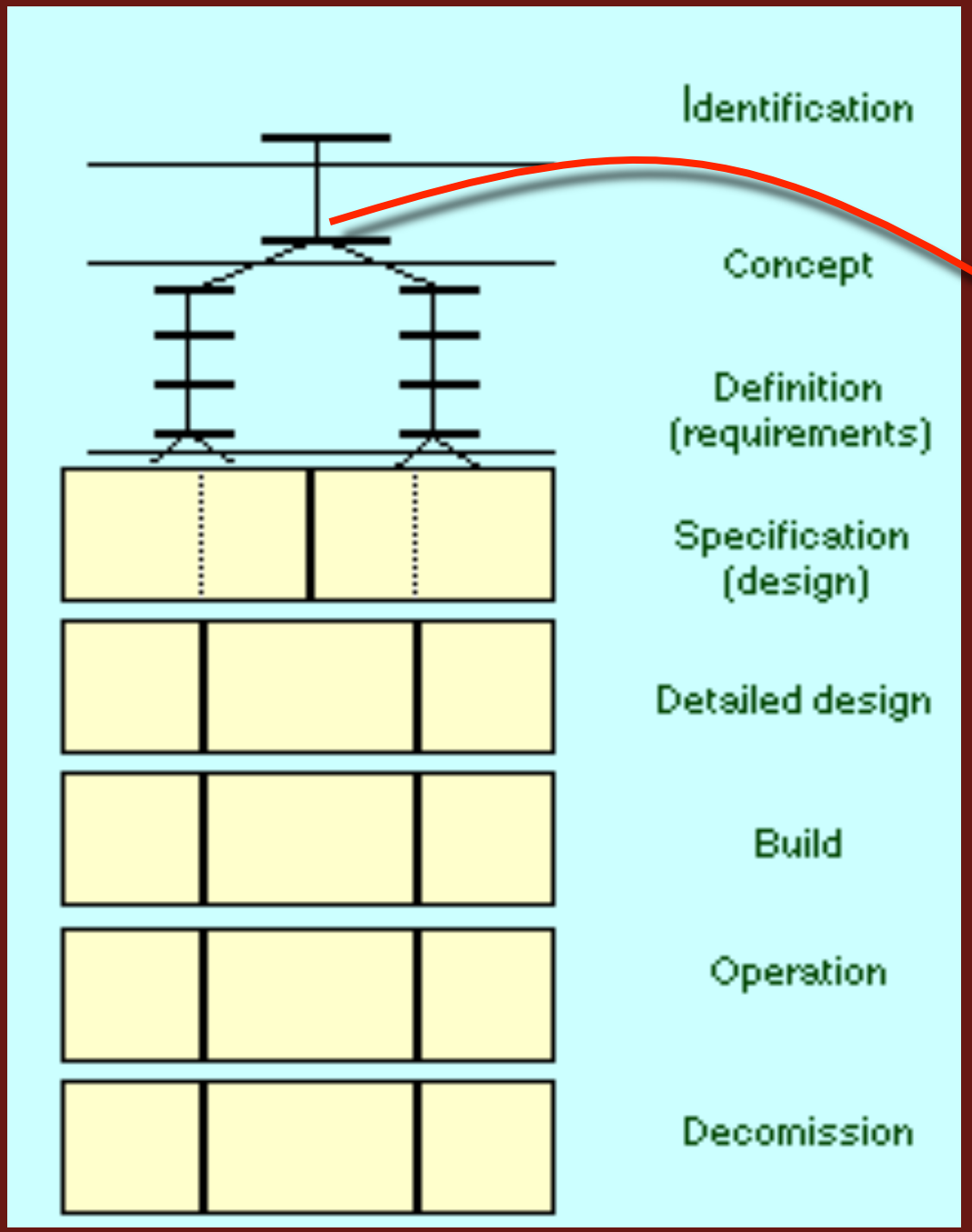
Identify the enterprise business entity (EBE):

- what it is, where it is,
- who the *stakeholders* are

- What is the identity of this entity
- What are the *Strategic relationships* to other entities
- *Goals* and *Objectives*

The above may be *shared* between several EBEs, or are appropriately instantiated. E.g., a goal of one EBE (e.g. a company goal) may become the objective of another EBE (become a project's objective) that decomposes this further (into project goals)

This is the description of the business / enterprise entity in its environment: what is its role and what are its relationships



Develop the Concept of the EBE. describing its

Business Model and Business Strategy

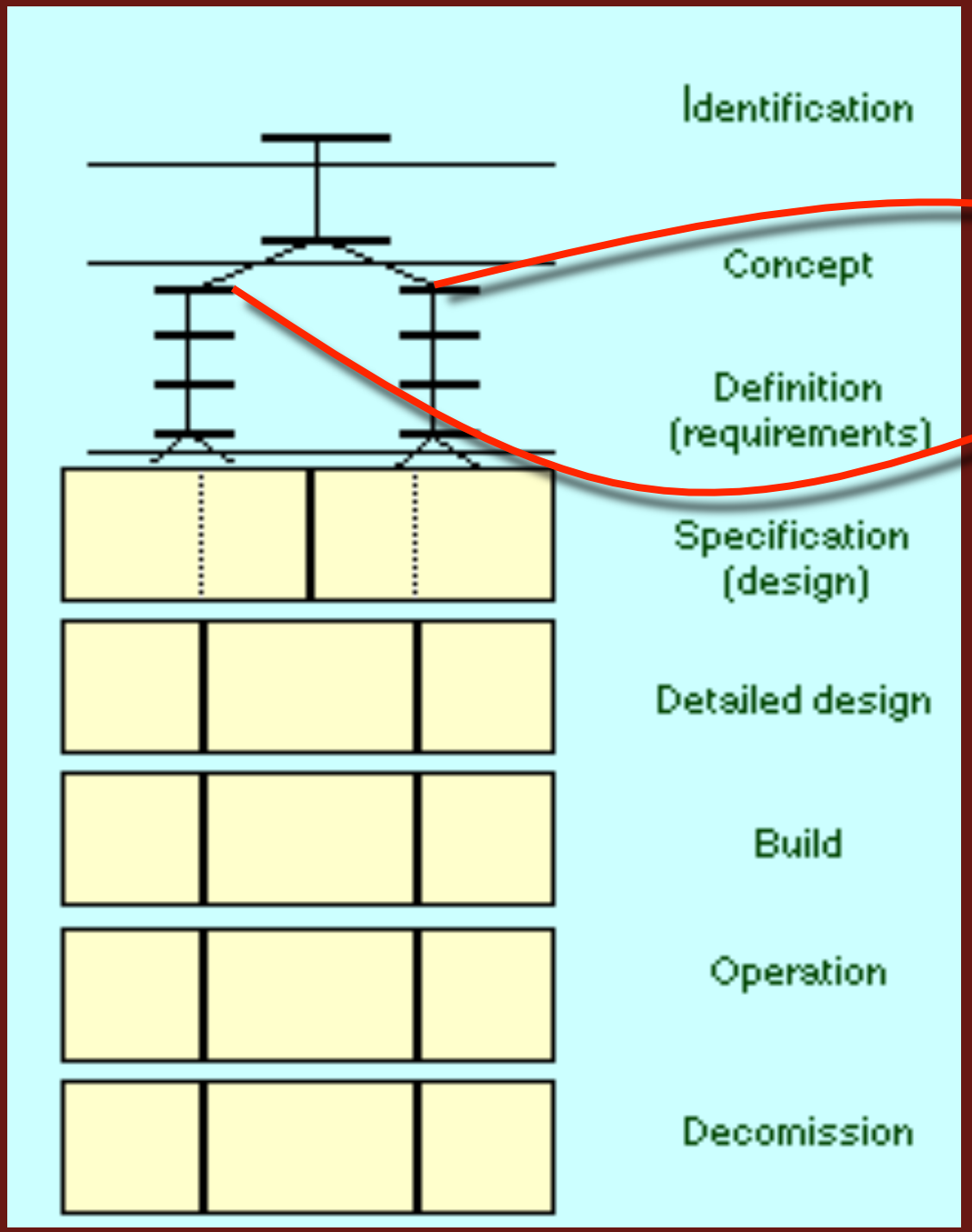
- Mission, Vision, Future
  Goals / Objectives
  why does this EBE exist?
  to deliver
  - what (product / service / value proposition)
  - when,
  - where,
  - to whom (customer) in what quality and quantity?
  - why?

- The EBE's *Capabilities* and *Competencies* of the EBS and what is its role in the *Value chain*

- Relationship to Programmes, Projects to satisfy the strategic objectives

- Values and Principles
  These *guide* the design and implementation as well as the operation of the EBE (many of these are shared by a set of EBEs)

This is the description of the 'Business Model' and Business Strategy as business people see it

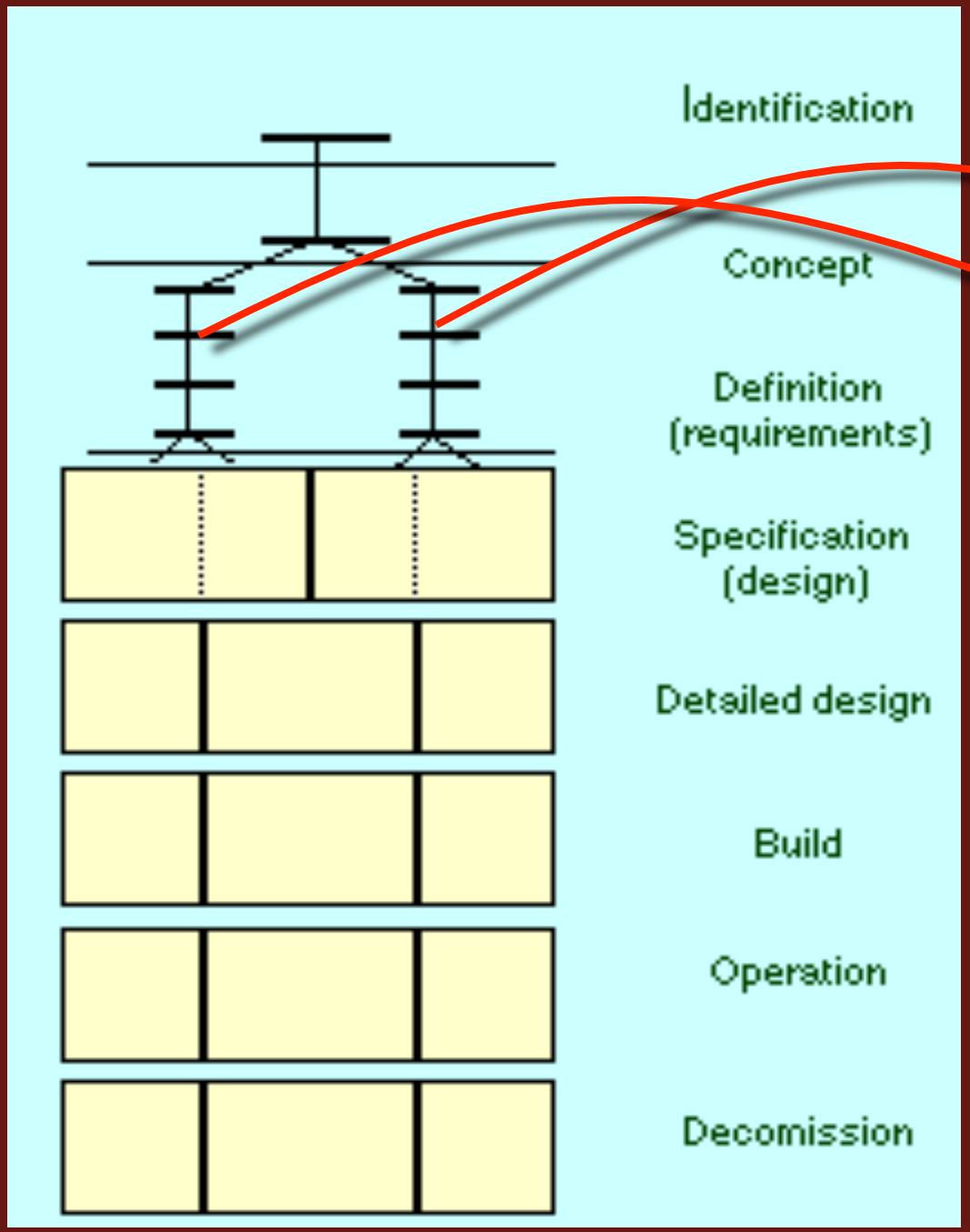


Define the requirements
to be satisfied by the EBE

- Production and Service Policies & Principles
- Management Policies & Principles

Note: this includes the human / organisational, business process, and technology oriented *policies & principles*

Many os these would already exist, but some new ones may be needed / old ones may need to be changed



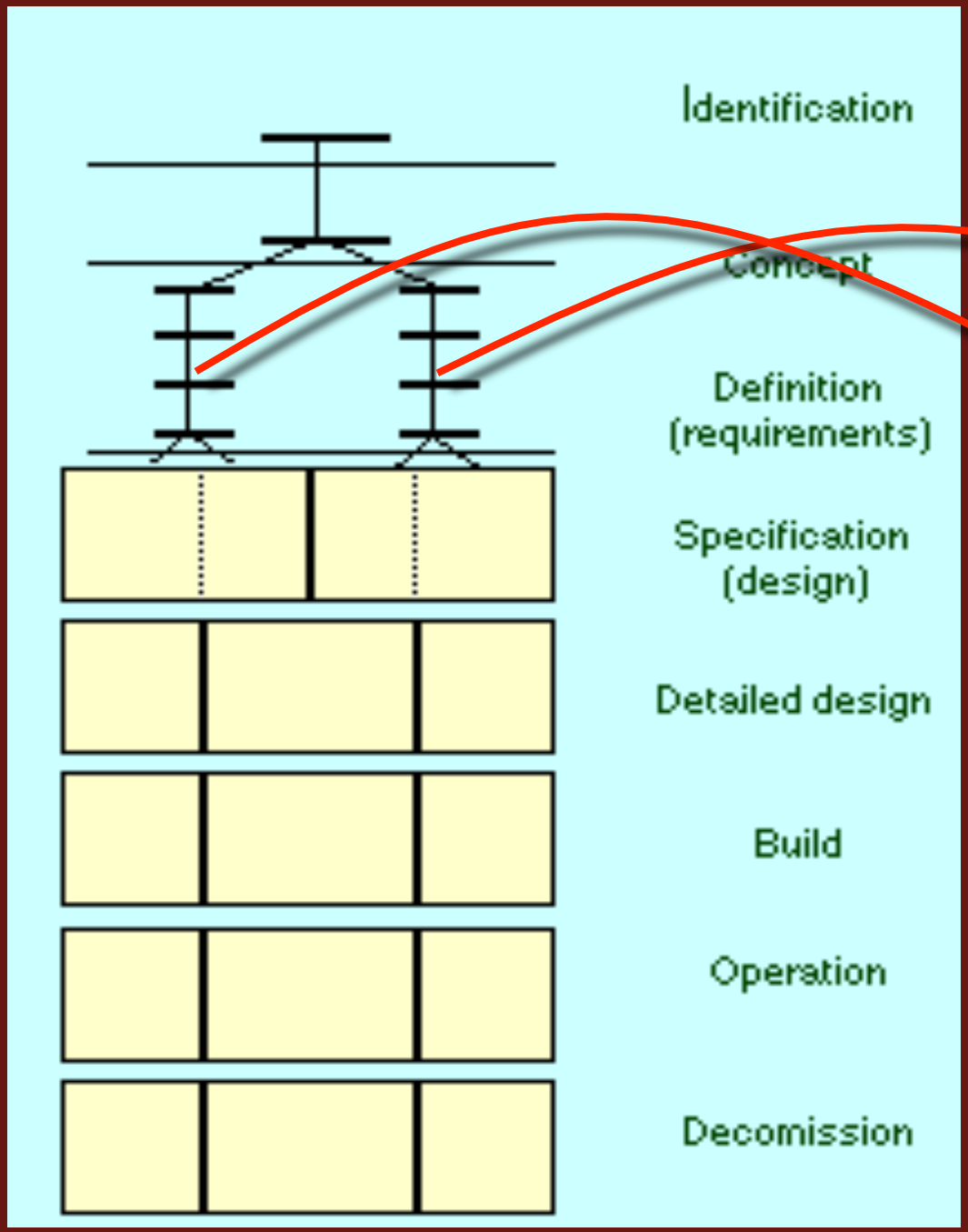
Define the requirements
to be satisfied by the EBE (cont'd)

Service Tasks

Management Tasks

Notice: we include
all tasks on this level -
(no matter whether the task
is intended to be performed by humans
or is to be automated)

Tasks can be defined as material or
information transformation activities
with inputs, outputs and controls

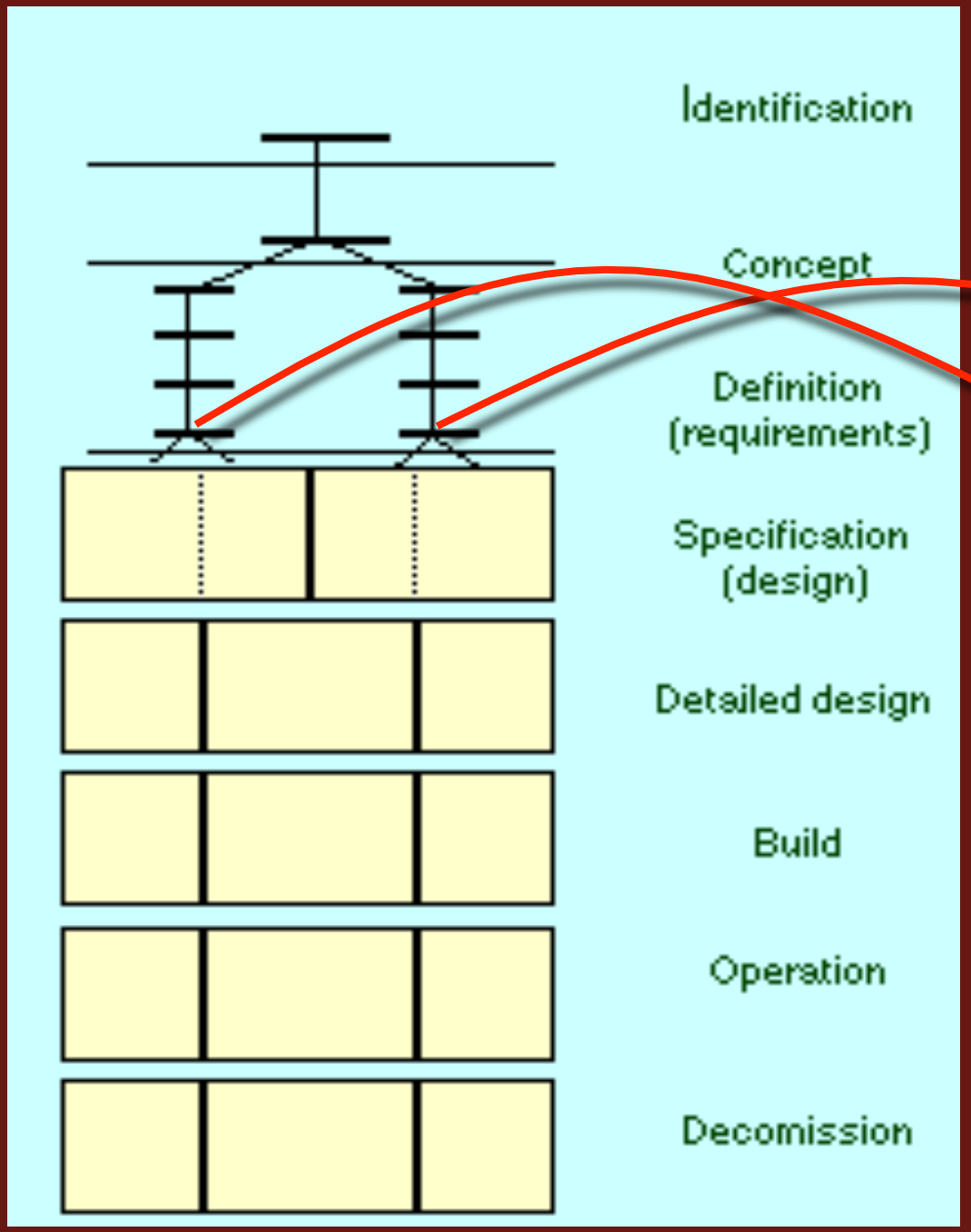


Define the requirements
to be satisfied by the EBE (cont'd)

Service
'task modules'

Management
'task modules'

('modules' are aggregate tasks that belong together, so these groups of tasks can be further defined in a more detailed way as a network of tasks)



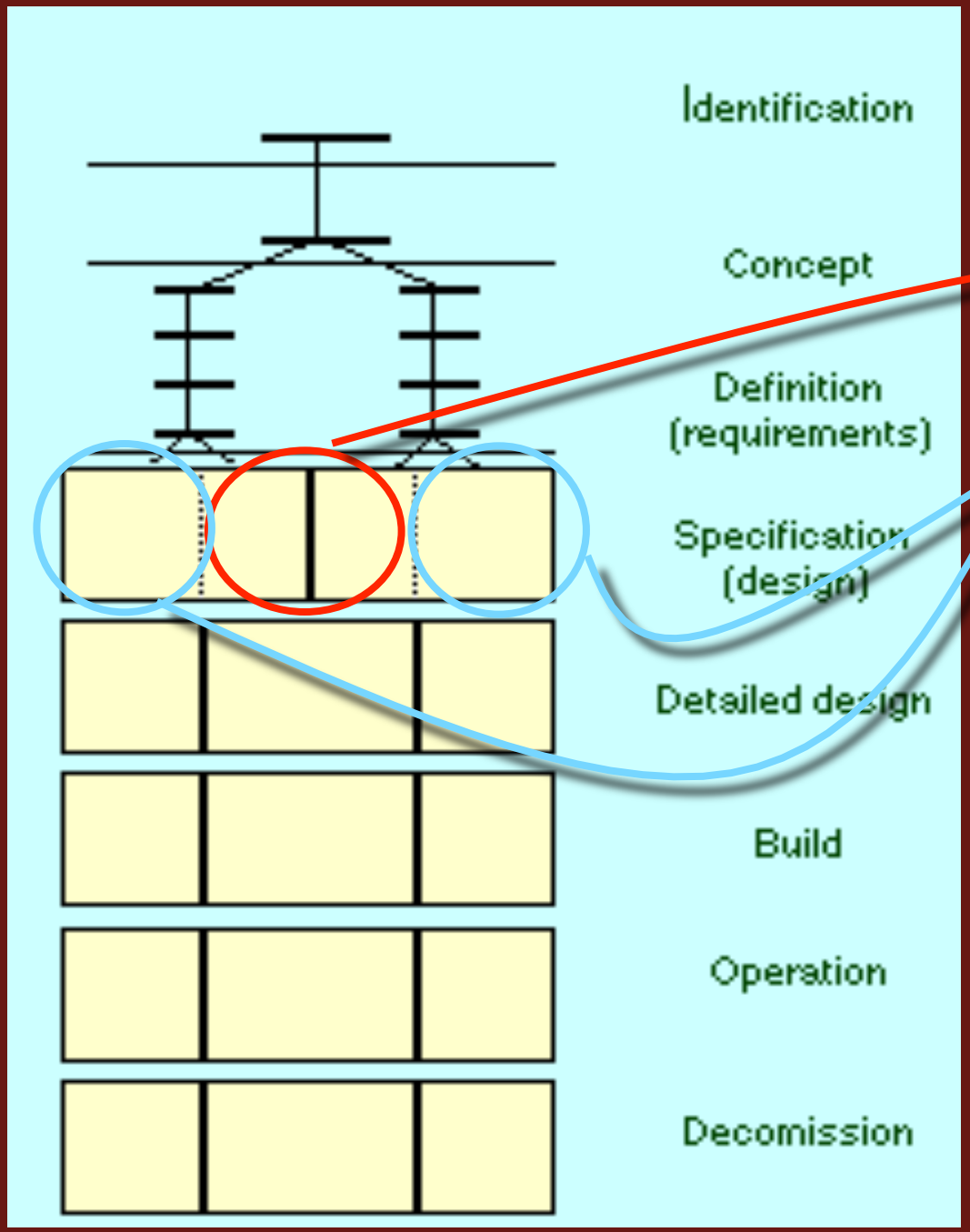
Define the requirements
to be satisfied by the EBE (cont'd)

Service Task Networks

Management Task Networks

(Define relationships between tasks such as information used, produced, and the interfaces between tasks. Usually expressed in some form of model, e.g., activity model, simulation model, process model, etc.)

This is usually referred to as the functional requirements specification of the EBE (with added list of non-functional requirements)

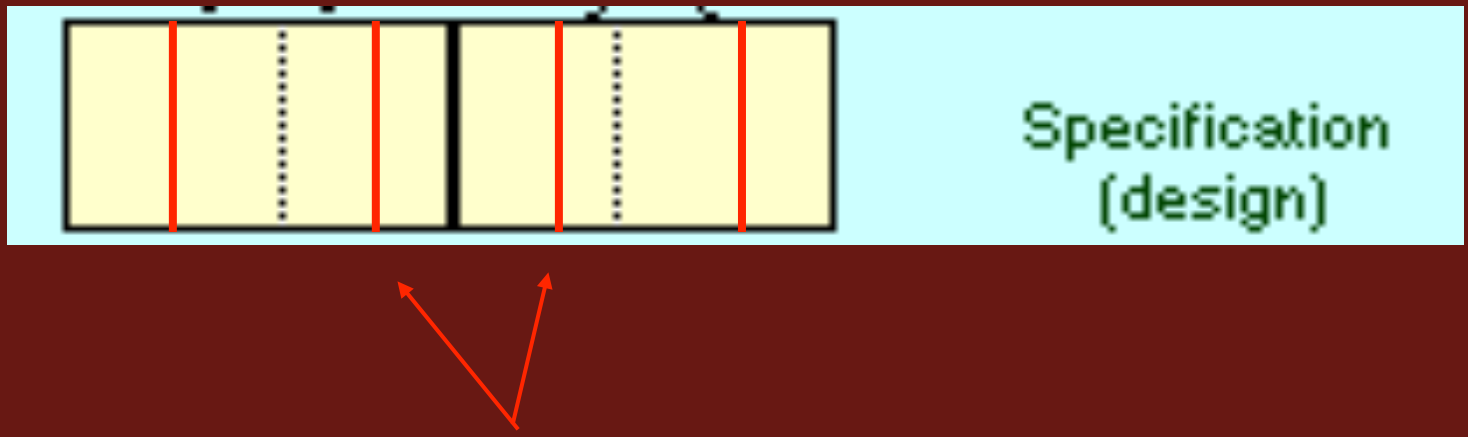


Specify the system design for the EBE

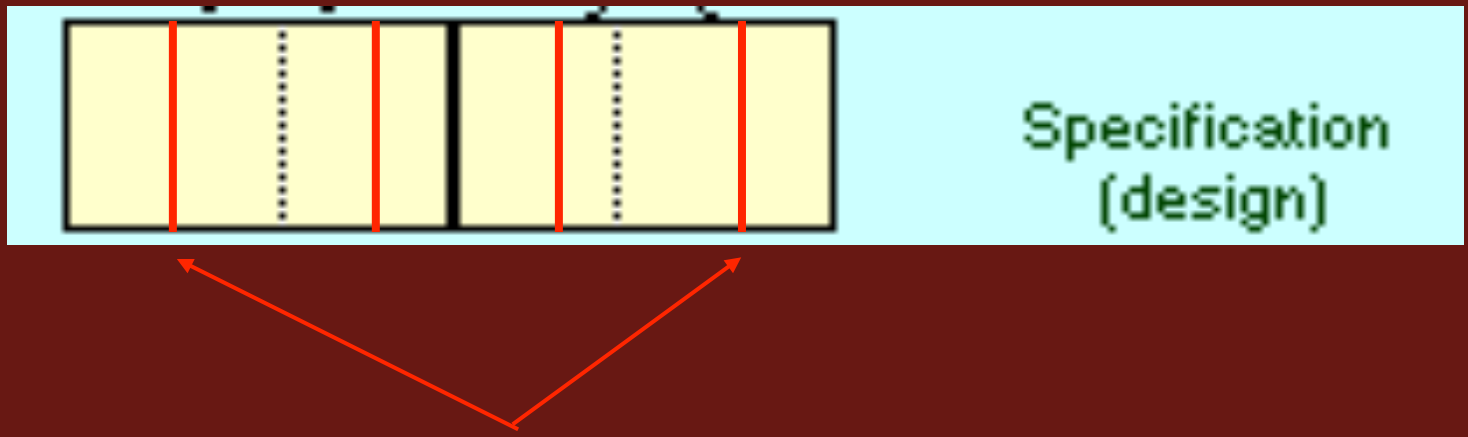
Automated vs Human tasks
in both the service & production and management & control of the EBE

Notice the dotted 'extent of automation line'

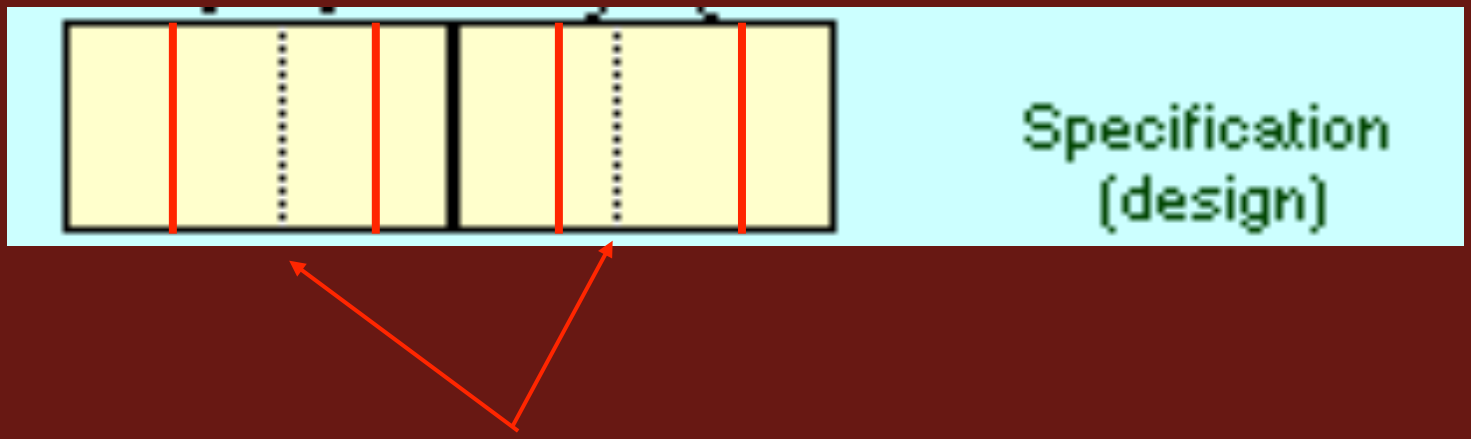
This is usually referred to as the architectural design (system design / preliminary design / high level design) of the EBE



‘Automatability’ line
(if everything that can be automated would be automated)

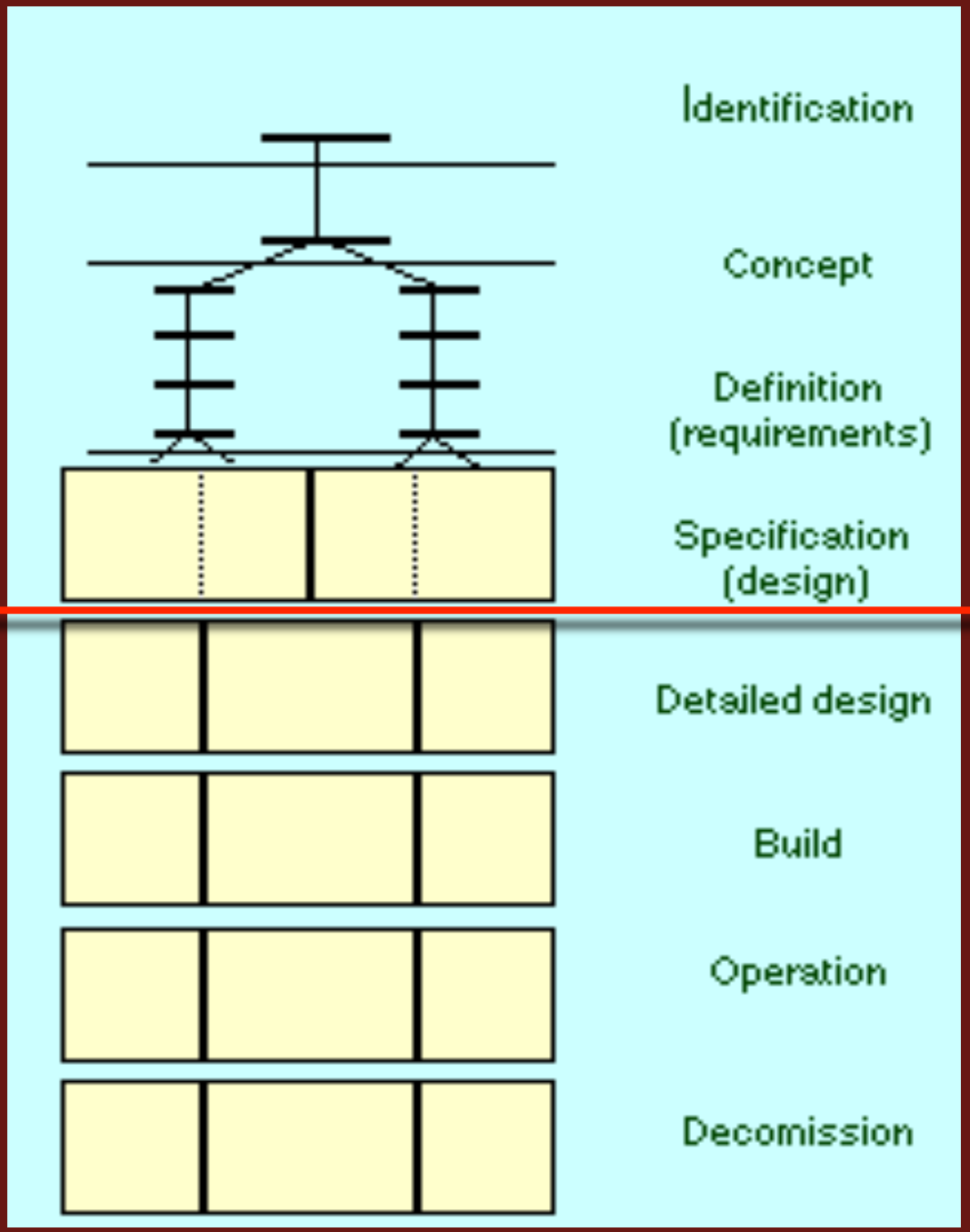


‘Humanisability’ line
(if everything that can be done by humans, would be)



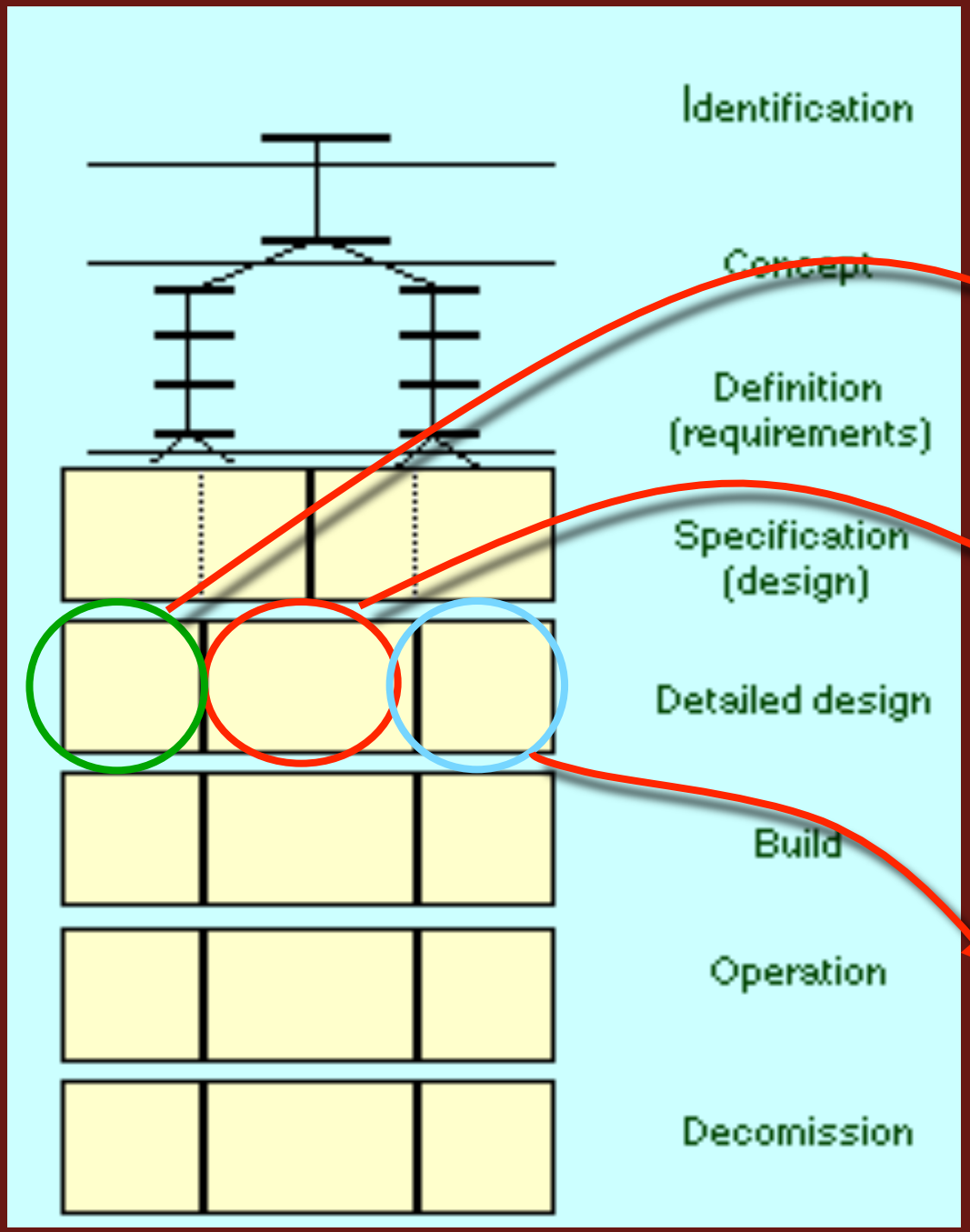
‘Extent of Automation’ line
(actual separation of human and automated tasks)

Important: the requirements definition (requirements specification) tolerates any choice between the two extremes, thus if interfaces are defined then later change (automation of human tasks, or the opposite - reverting to human instead of automated execution) is straightforward



At this point we have a *Master Plan (or 'Architectural Design')* of the (to be) EBE, which can be implemented in one go or in co-ordinated steps in stages

The implementation plan can be incorporated in (added to) the *Business Plan*, and in case of an existing EBE this includes a *transition plan*

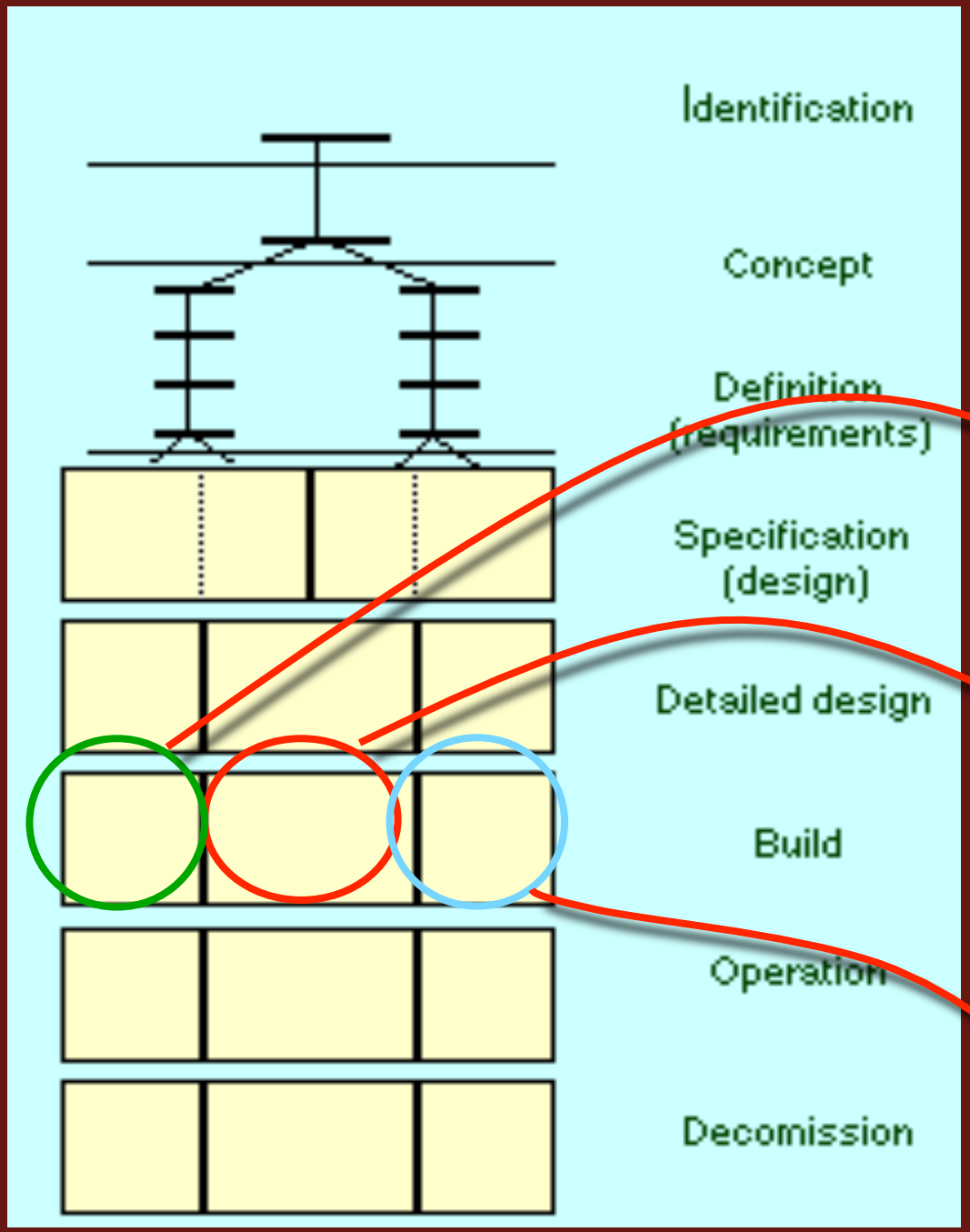


Detailed design of the EBE
(may be done in parallel projects
usually performed by designers of
different specialisation)

Design the *equipment (software & hardware) for production and service delivery*

Design the *human organisation*
(task- and job descriptions,
instruction manuals, training
needs, hiring guidelines, etc.)

Design the *management and control system software*: ERP, MIS (applications, database management systems, communications, secutiry) and *hardware*: (controllers, sensors, processing, storage & network infrastruture…)



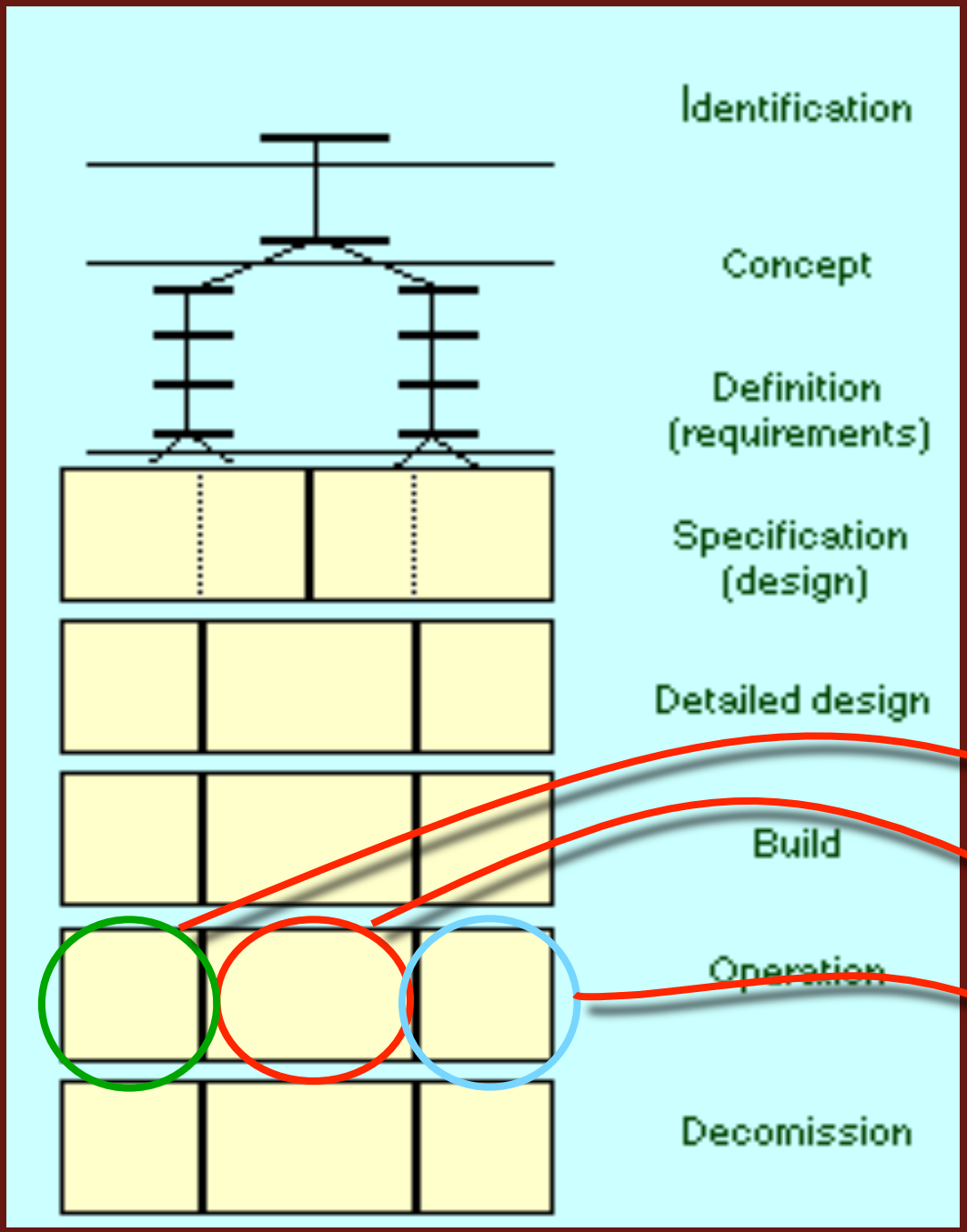
Build

(commission, procure, construct, deploy, install, configure, ...) the three subsystems

production system

human organisation

management and control system

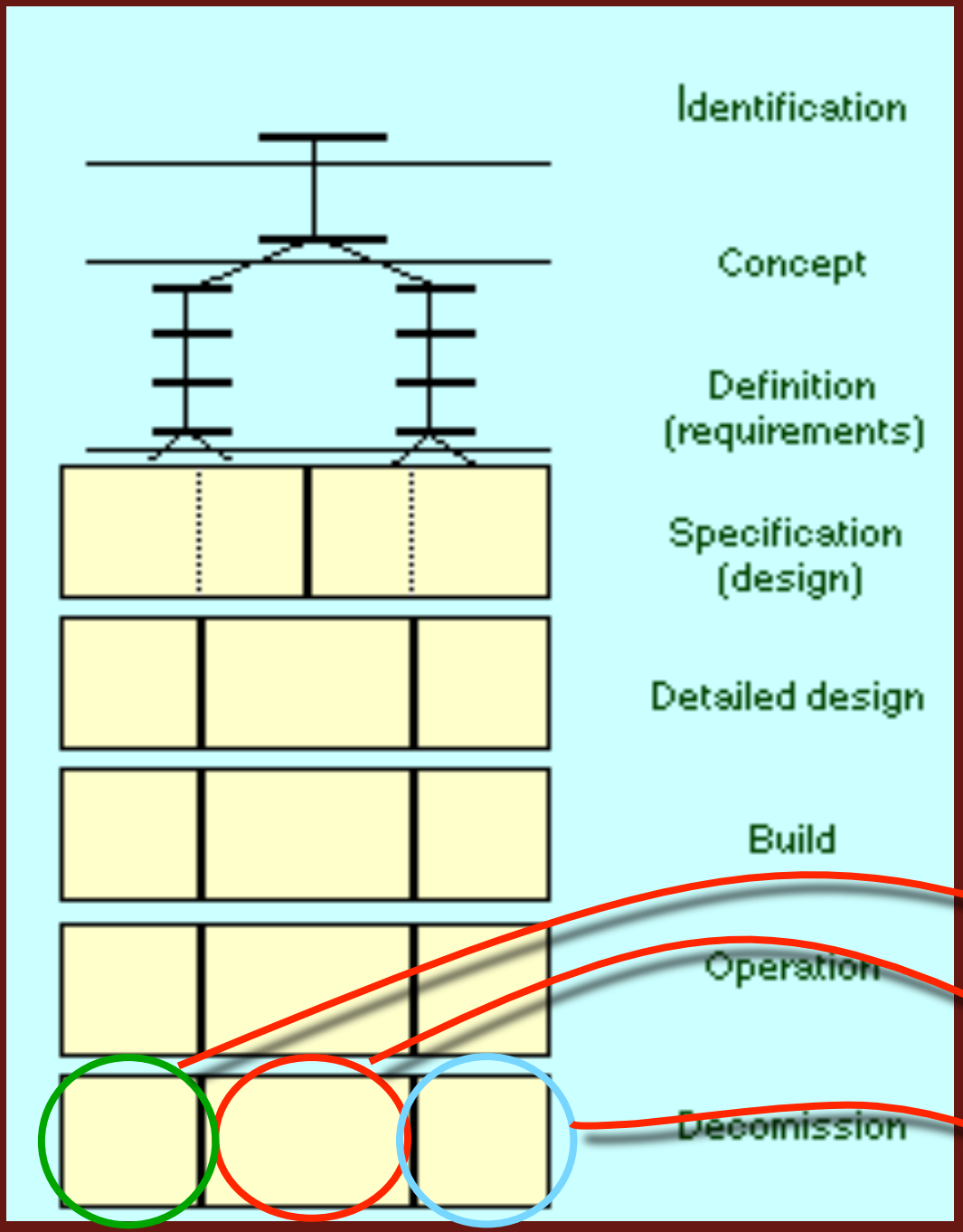


Operate the EBE

production system

human organisation

management and control system

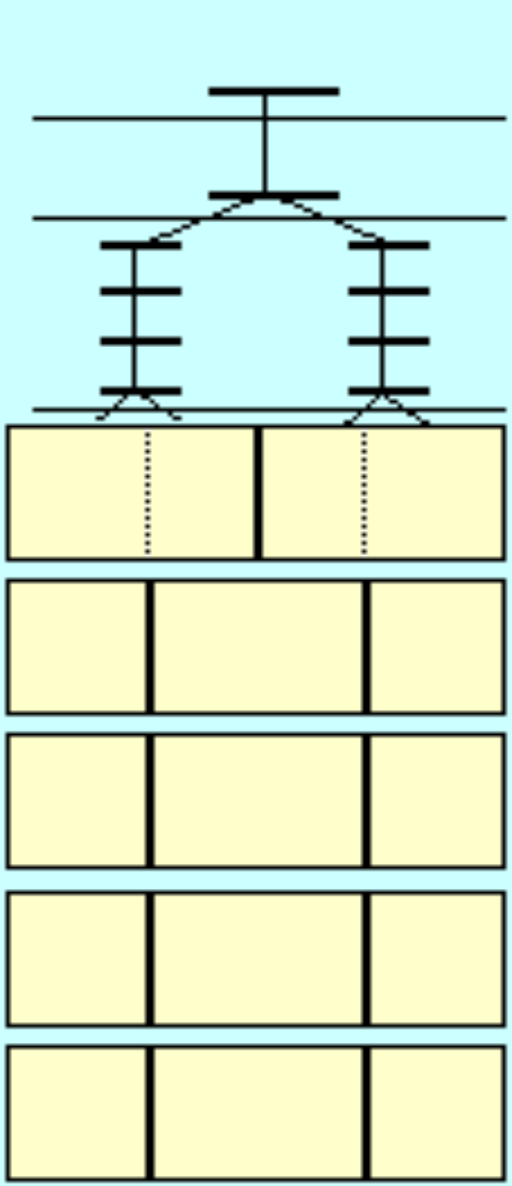


Decommission all or part of the three subsystems

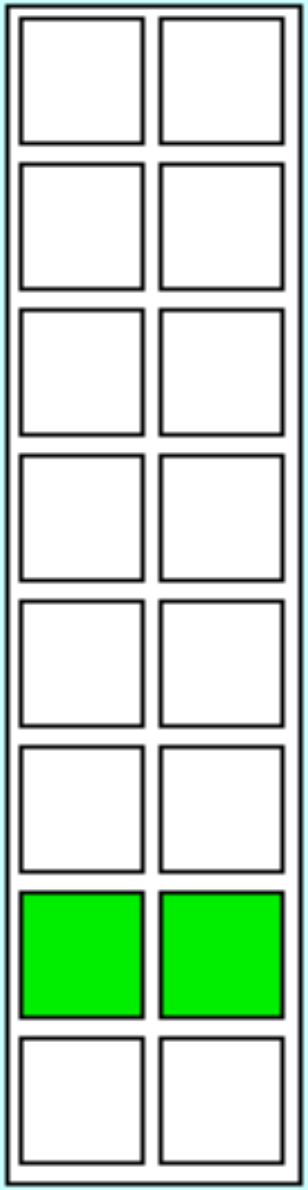
production system

human organisation

management and control system



Identification
Concept
Definition (requirements)
Specification (design)
Detailed design
Build
Operation
Decomission



The representation of an EBE's life cycle using PERA's graphical notation

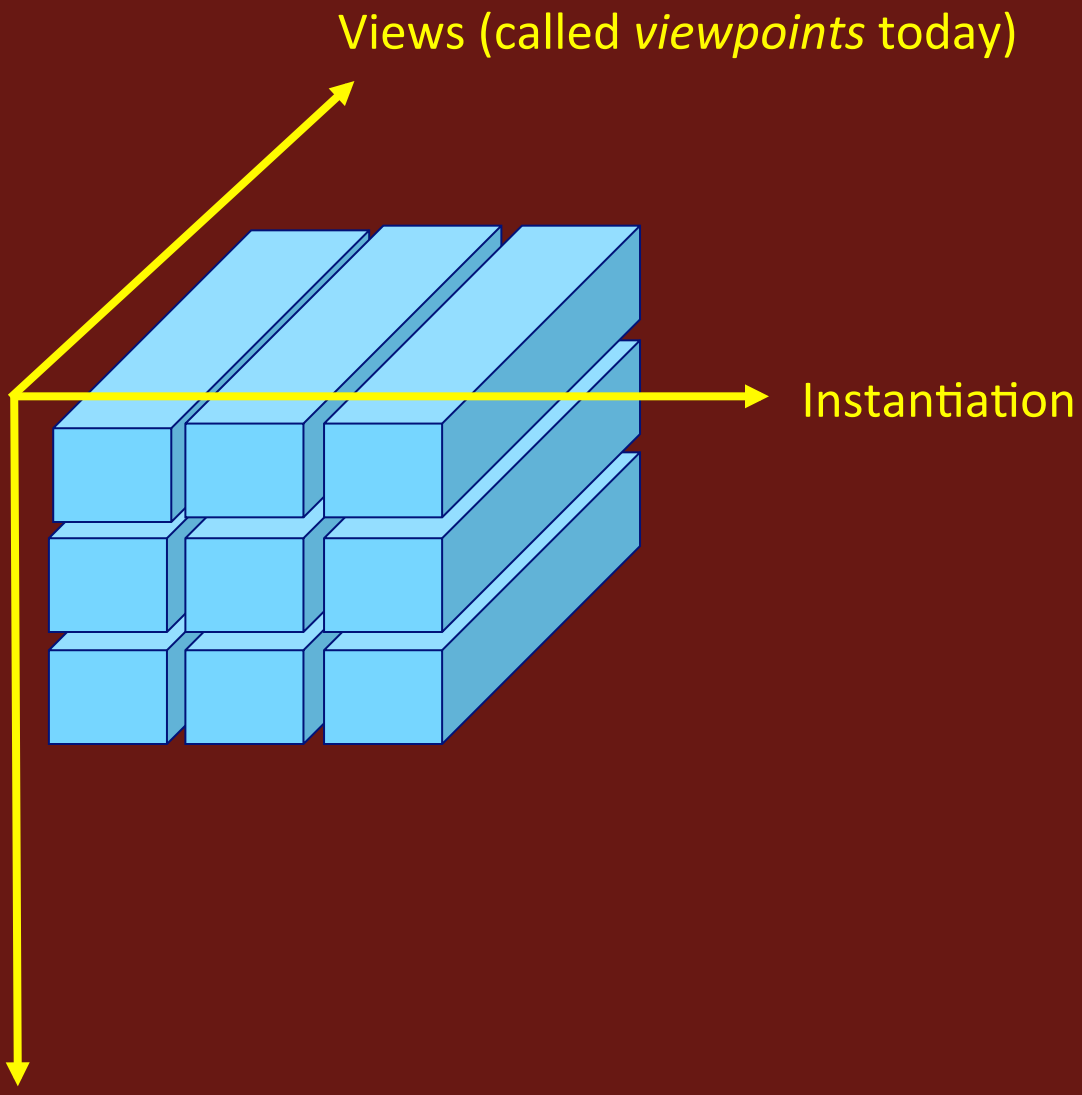
A shorthand graphical notation to represent the life cycle of an EBE ('chocolate bar')

CIMOSA

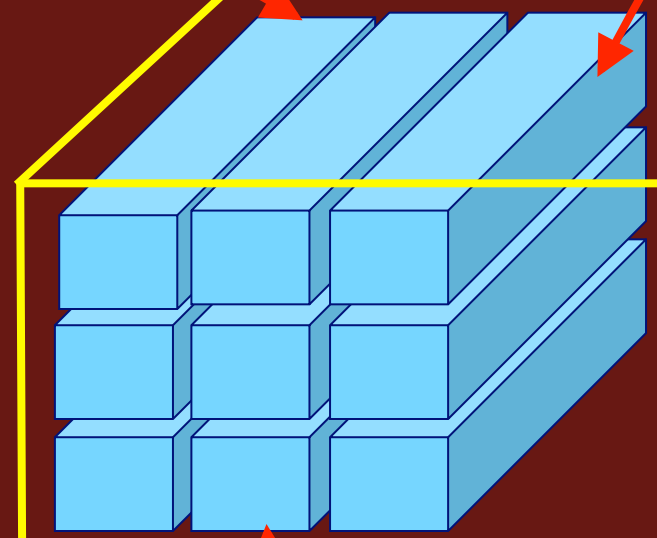
- Developed by the AMICE Consortium (originally for CIM systems design and implementation)

  We only introduce part of CIMOSA here (i.e. CIMOSA's modelling framework) to illustrate some important concepts that influenced the development of several AFs

- CIMOSA was used in the manufacturing/automotive industry in Europe



the CIMOSA cube introduced a classification of enterprise descriptions



Generic models (so-called '*ontological models*')

Views (called *viewpoints* today)

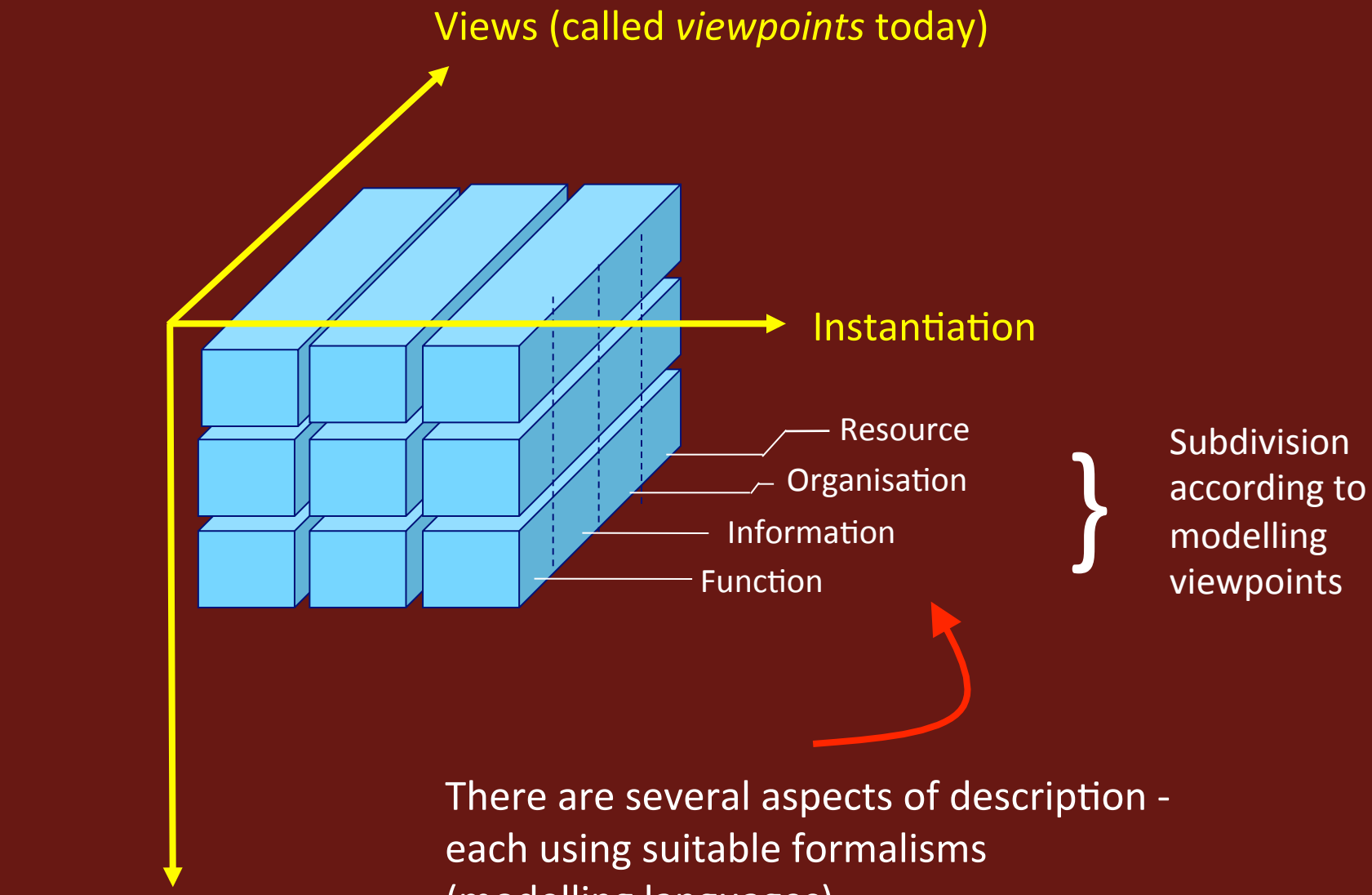
Descriptions of the individual enterprise ('*particular models*')

Instantiation

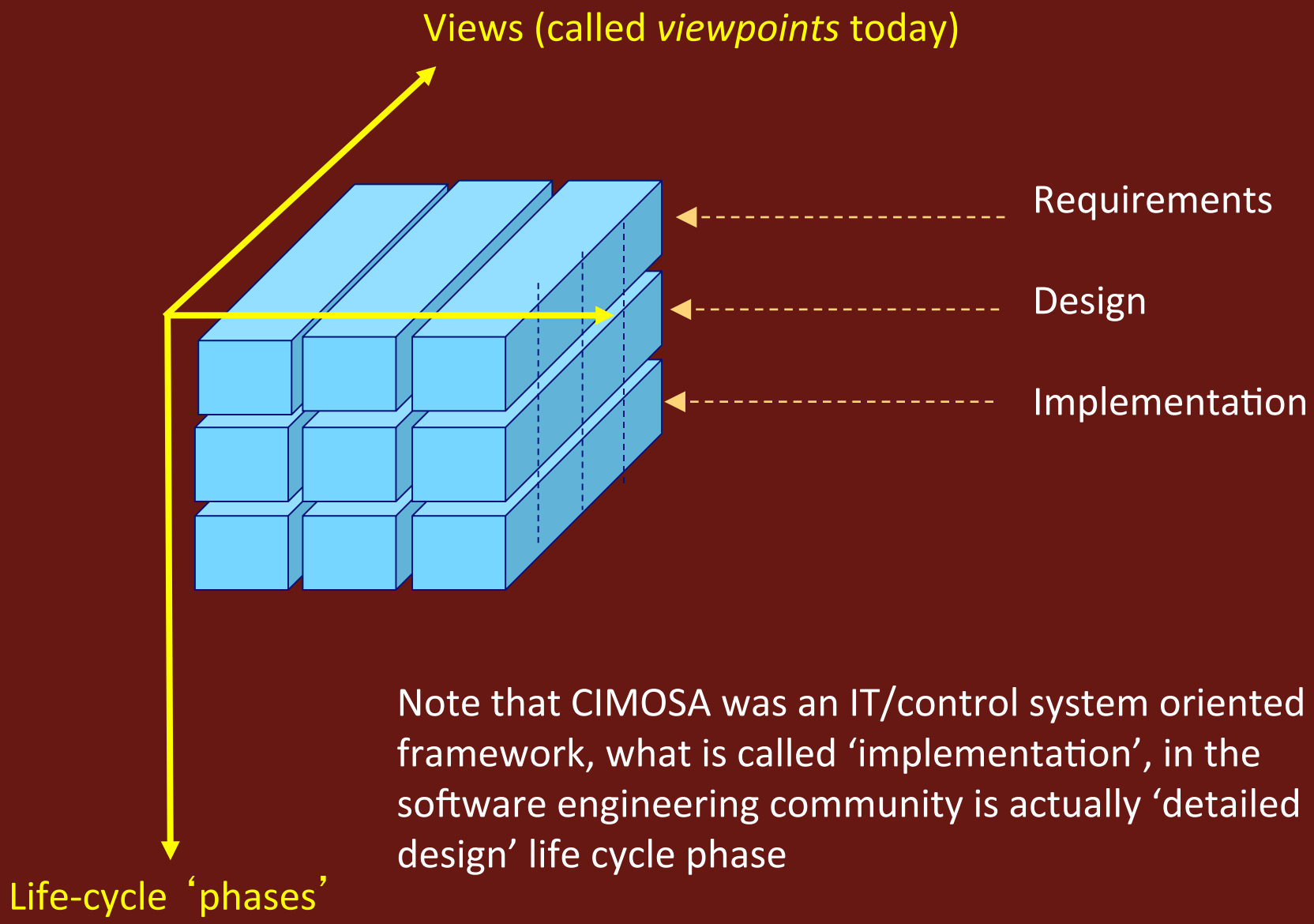
typical enterprise models or *reference models* (CIMOSA calls these '*partial models*')

Life-cycle 'phases'

The Instantiation dimension



The view (viewpoint) dimension

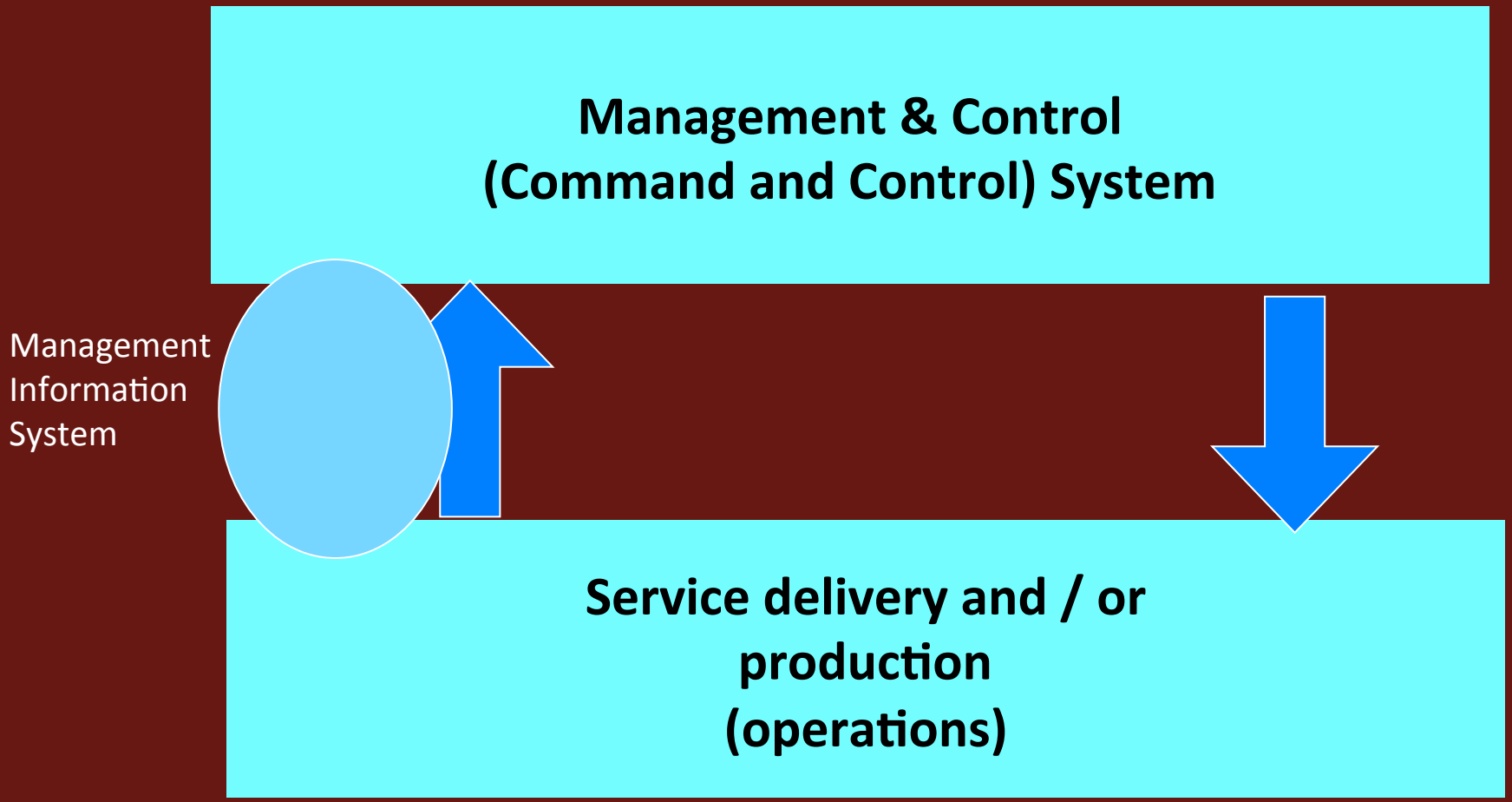


Note that CIMOSA was an IT/control system oriented framework, what is called 'implementation', in the software engineering community is actually 'detailed design' life cycle phase

Life cycle dimension

GRAI GIM

- Developed by the GRAI Laboratory (U.of Bordeaux, Prof G.Doumeingts) originally for the design of production management systems
- Extensive use in manufacturing, service, for BPR, performace indicator development and benchmarking (mainly in Europe)
- We only study in this course *one aspect* of GRAI-GIM, namely its unique (and very generic / generally useful) reference model

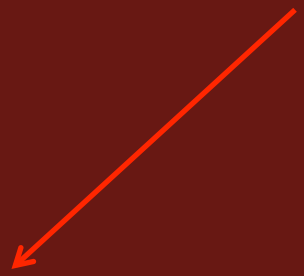


GRAI GIM Reference Model (this will be relevant later when we discuss how to model the management and control system; we do not discuss other aspects of GRAI GIM here)

Zachman framework

- Developed by J. Zachman (IBM) for enterprise description
- Similar to CIMOSA, but has more detailed classification of viewpoint descriptions
- No explicit definition of generic and partial models
- Often used in business and government sectors

Life-cycle activities correspond to types of people involved in an enterprise architecture project (the planner strategy maker / owner, the architect, the designer, the builder / subcontractor, user)



Strategy

Analysis

Design

Construction

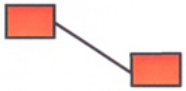
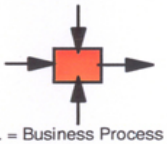
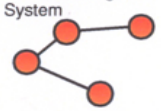
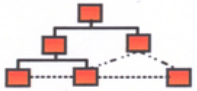
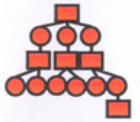
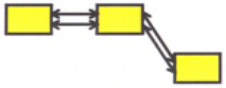
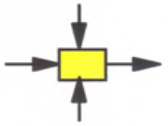
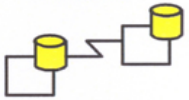
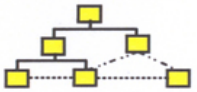
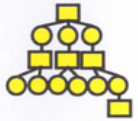
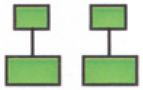
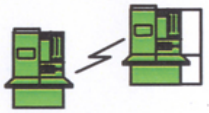
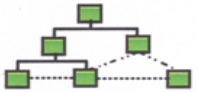
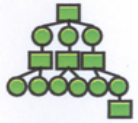
Documentation

Production

	Data (What)	Function (How)	Network (Where)	People (Who)	Time (When)	Motivation (Why)
Strategy						
Analysis						
Design						
Construction						
Documentation						
Production						

Descriptions from various aspects
each using suitable formalisms
(modelling languages) not predetermined
in the framework

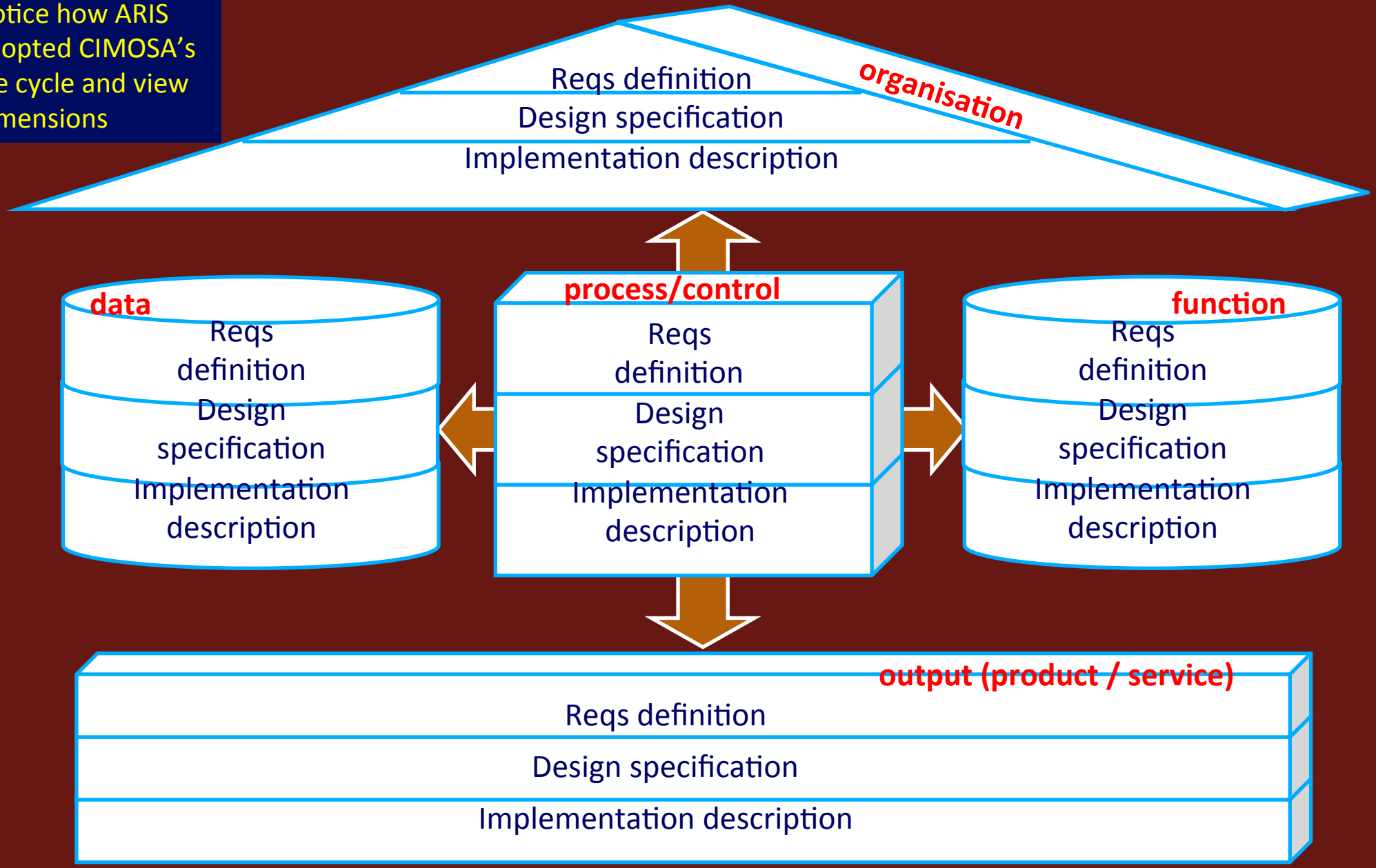
ENTERPRISE ARCHITECTURE - A FRAMEWORK ™

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business  ENTITY = Class of Business Thing	List of Processes the Business Performs  Process = Class of Business Process	List of Locations in which the Business Operates  Node = Major Business Location	List of Organizations Important to the Business  People = Major Organization Unit	List of Events/Cycles Significant to the Business  Time = Major Business Event/Cycle	List of Business Goals/Stratgies  Ends/Means = Major Business Goal/Strategy	SCOPE (CONTEXTUAL) *Planner*
BUSINESS MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model  Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model  Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System  Node = Business Location Link = Business Linkage	e.g. Work Flow Model  People = Organization Unit Work = Work Product	e.g. Master Schedule  Time = Business Event Cycle = Business Cycle	e.g. Business Plan  End = Business Objective Means = Business Strategy	BUSINESS MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model  Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture  Proc .= Application Function I/O = User Views	e.g. Distributed System Architecture  Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture  People = Role Work = Deliverable	e.g. Processing Structure  Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model  End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model  Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design  Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture  Node = Hardware/Systems Software Link = Line Specifications	e.g. Presentation Architecture  People = User Work = Screen Format	e.g. Control Structure  Time = Execute Cycle = Component Cycle	e.g. Rule Design  End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition  Ent = Field Reln = Address	e.g. Program  Proc.= Language Statement I/O = Control Block	e.g. Network Architecture  Node = Address Link = Protocol	e.g. Security Architecture  People = Identity Work = Job	e.g. Timing Definition  Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification  End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

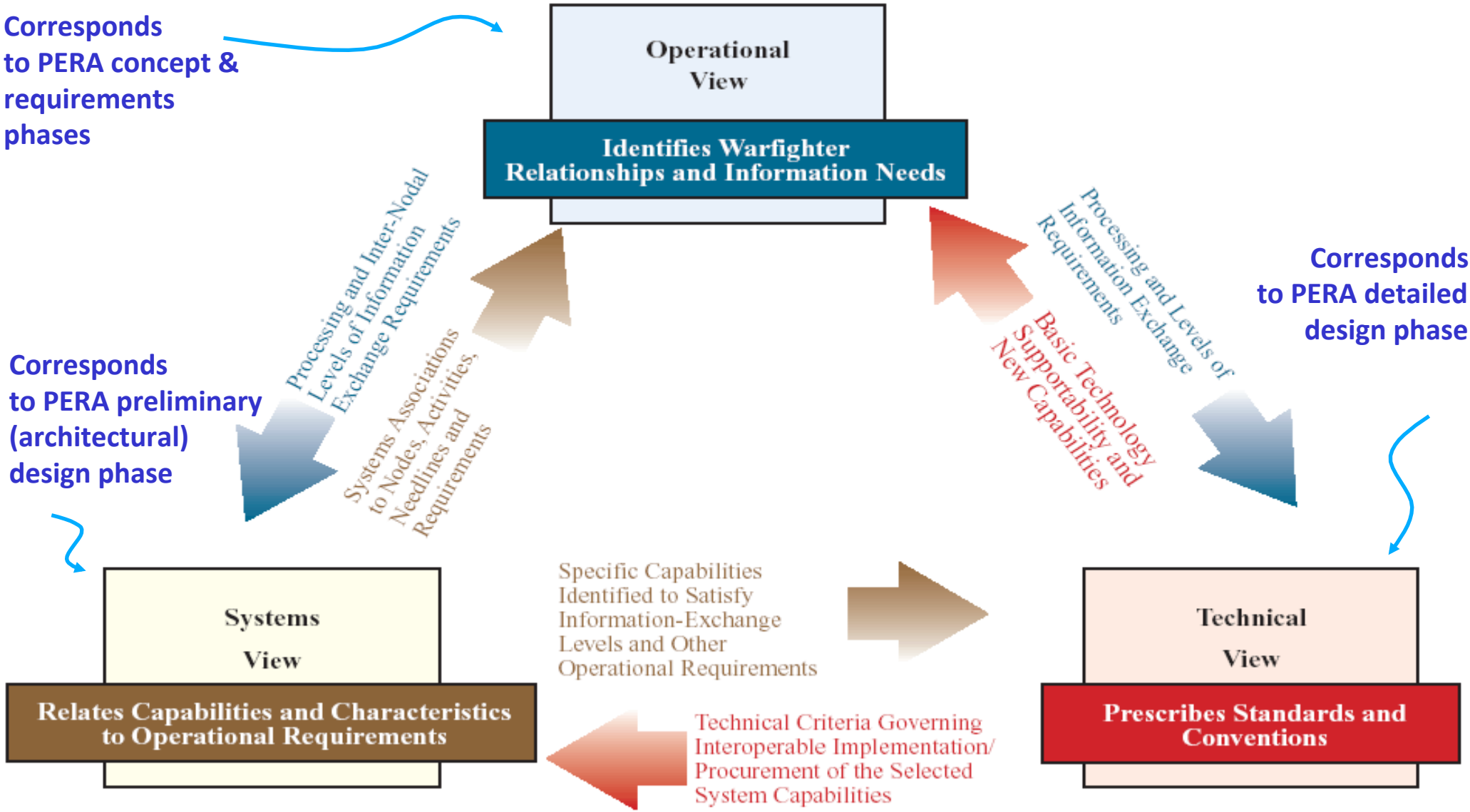
John A. Zachman, Zachman International

ARIS (ARchitecture of Information Systems) [IDS Scheer]

Notice how ARIS adopted CIMOSA's life cycle and view dimensions



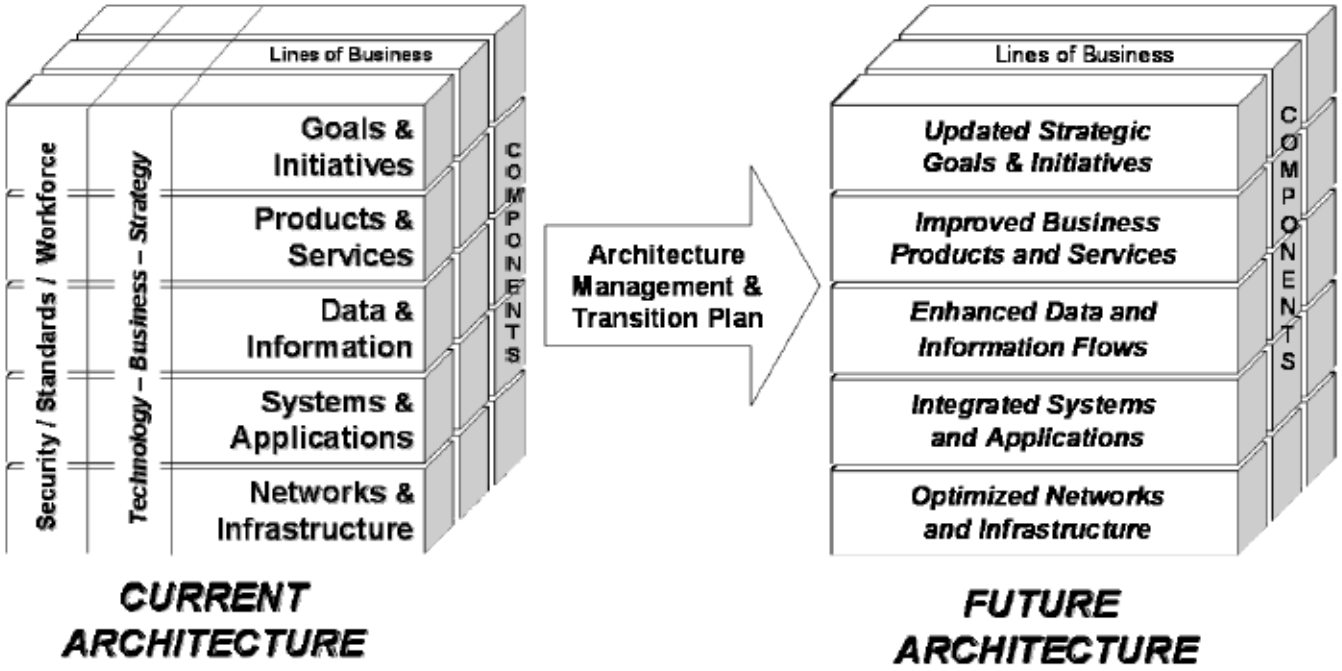
C4ISR / DoDAF (US Department of Defense Architecture Framework)



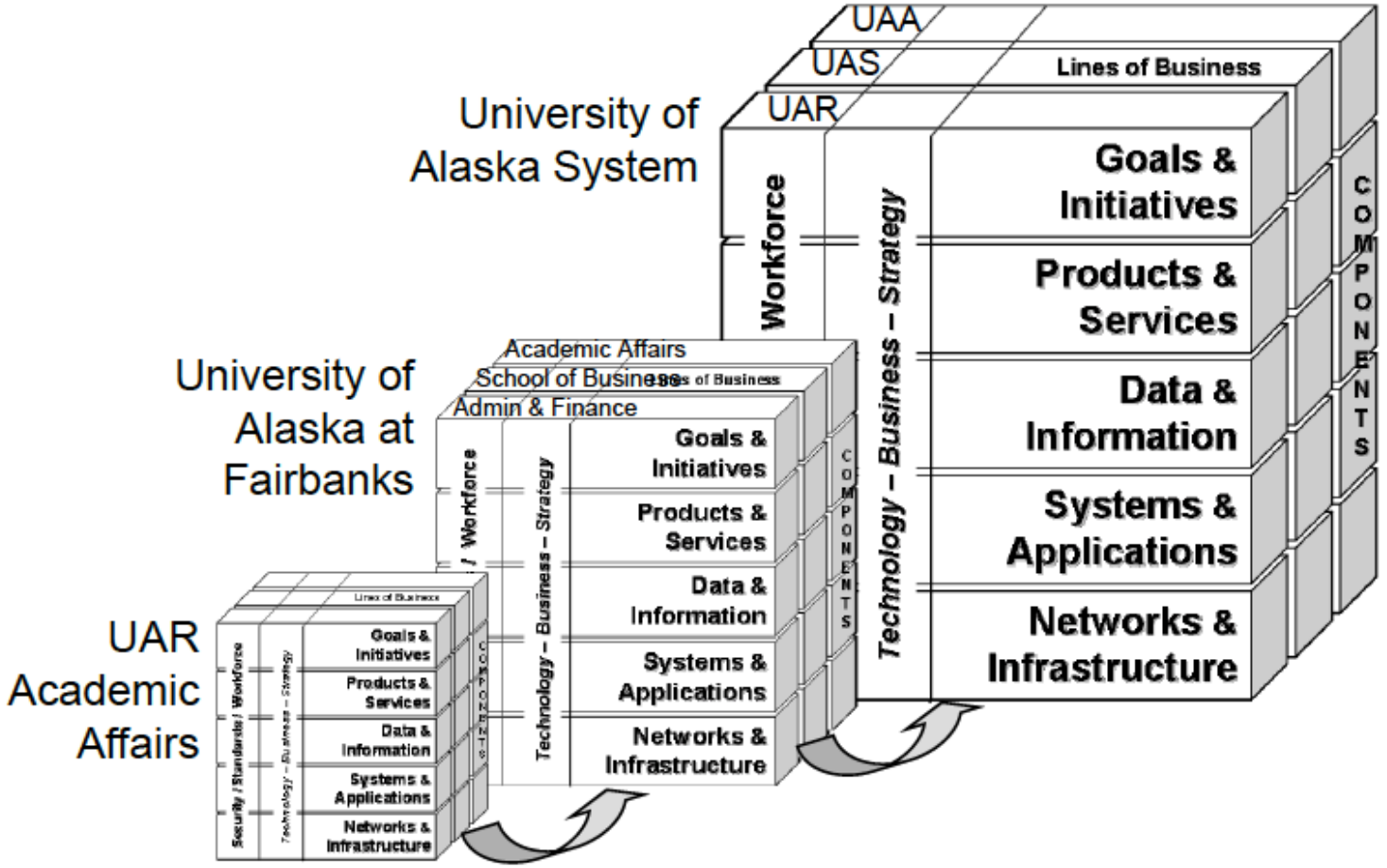
Enterprise Architecture - Defined

Enterprise Architecture: The analysis and documentation of an enterprise in its current and future states from a strategy, business, and technology perspective. ***EA = S + B + T***

An Introduction to Enterprise Architecture © 2005



Enterprise Architecture:
A Scalable Model of an Organization



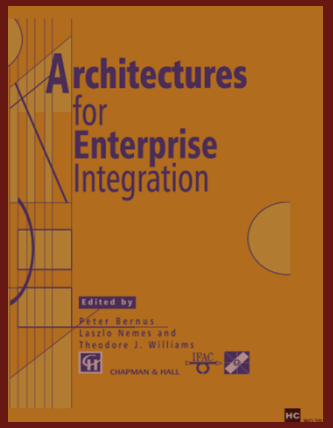
The GERAM Enterprise Architecture Framework
is a *generalisation* of these various frameworks and was developed by the IFIP/IFAC Task Force on Enterprise Integration (1990-2001)

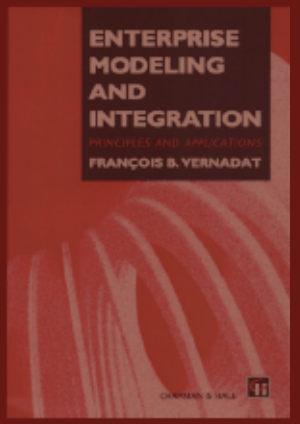
- ISO Standard 15704:2000 ; 2006 is based on GERAM and defines the Requirements for Enterprise Reference Architectures (called today 'Architecture Frameworks')

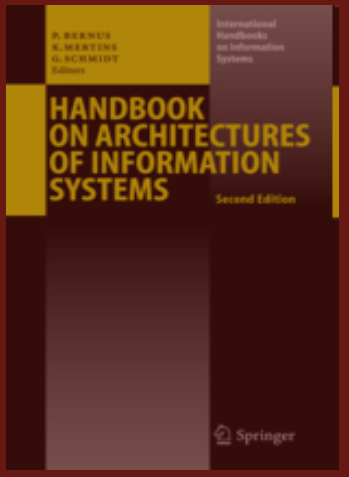
  Annex A of ISO15704:2000 ; 2006 is the detailed description of GERAM

- Also see GERAM on the web at www.ict.griffith.edu.au/~bernus

  [Note that the WIKIPEDIA entry is outdated]







EA as a Global Standard

- **International Standards for EA have existed for over 10 years:**
  - **ISO 14258 (1998):** Industrial AIS - Concepts and Rules for Enterprise Models
  - **ISO 15704 (2000):** Requirements for Enterprise Reference Architectures & Methods
  - **CEN ENV 40003 (1991):** CIMOSA Architecture Framework. Pre EN ISO 19439 (2002)
  - **CEN ENV 12204 (1996):** Constructs for Enterprise Modeling. Pre EN ISO 19440 (1996)
- **Specific approaches to EA have existed for over 15 years:**
  - **Private Sector**
    - **Open Standard Approaches**
      - Zachman EA Framework
      - Spewak EA Planning Method (EAP)
      - The Open Group Architecture Framework (TOGAF)
      - EA3 Cube Framework
    - **Proprietary Approaches**
      - Consulting Firms (e.g., Gartner, Meta Group)
      - Technology Firms (e.g., IBM, Microsoft)
  - **Public Sector**
      - Federal EA Framework & Reference Models (FEAF/FEA-RMs)
      - Department of Defense Framework (DODAF)
      - National Association of State CIO's EA Toolkit

What is the purpose of GERAM / ISO15704?

- GERAM organises enterprise integration / enterprise architecture knowledge
- It considers all components and aspects of the enterprise, including human and technical
- It applies to any complex entity (enterprises, networks of enterprises, projects, complex products etc., in any field of industry, service or government)
- GERAM is a generalisation of all Architecture Frameworks

The Components of GERAM (foundational concepts)

The operational enterprise
('Enterprise Entity')

EOSs

The Components of GERAM

We build models for various purposes, such as to express a design, to experiment with or to analyse a design, to represent an existing system, to support communication about the existing or future state of a system, to support decision making, etc. A *model* of a system is (for the purposes of an investigation) equivalent to the system

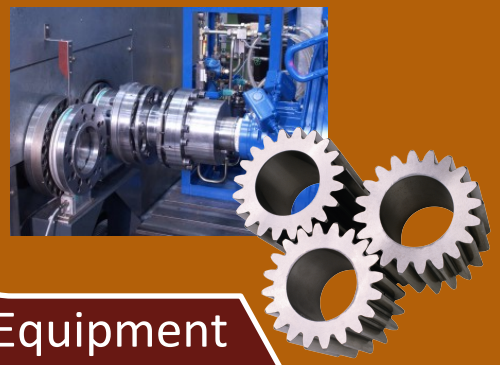
Enterprise models

EMs

EOSs

The Components of GERAM

These are the actual components out of which the Enterprise Entity is built



Equipment
SW/HW

Humans

SW/HW

EMs

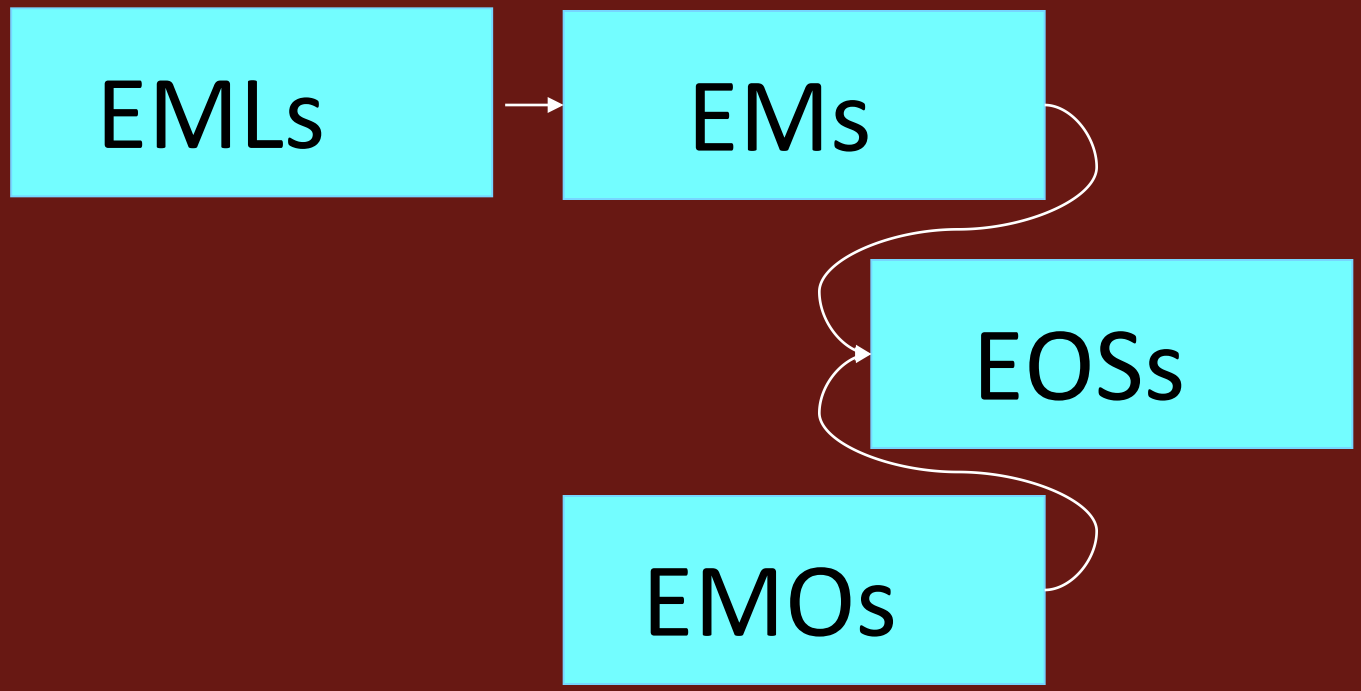
EOSs

EMOs

Enterprise ‘modules’

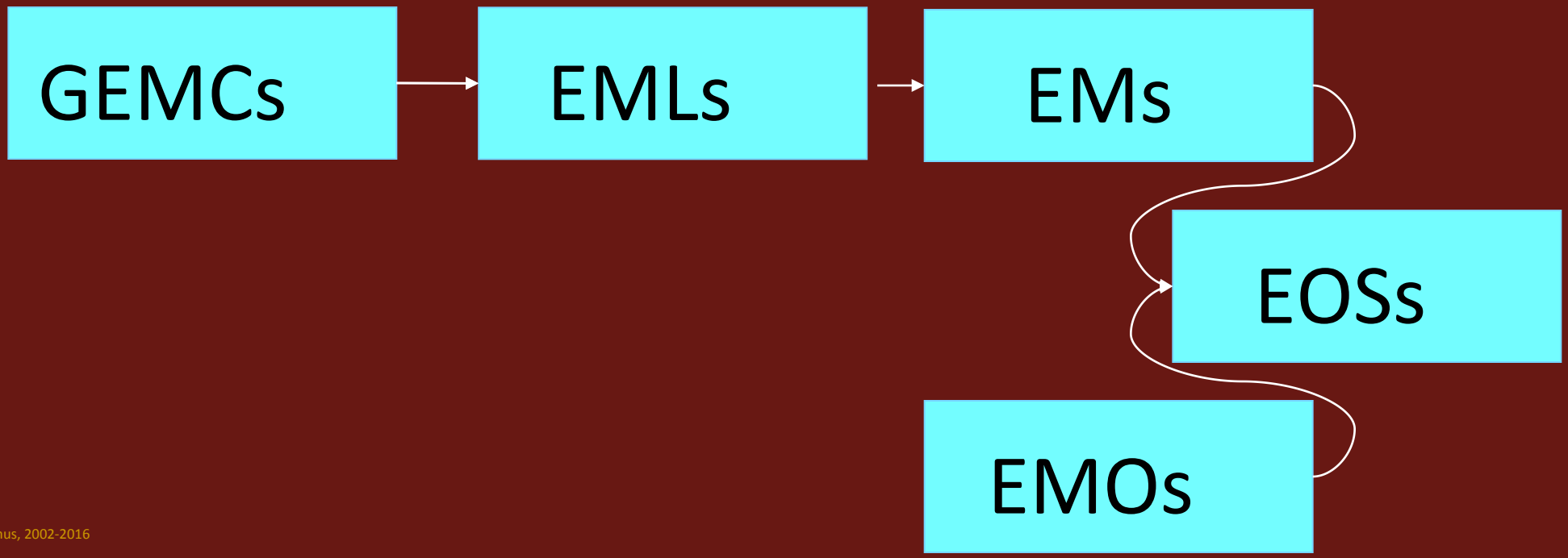
The Components of GERAM

Enterprise modelling languages

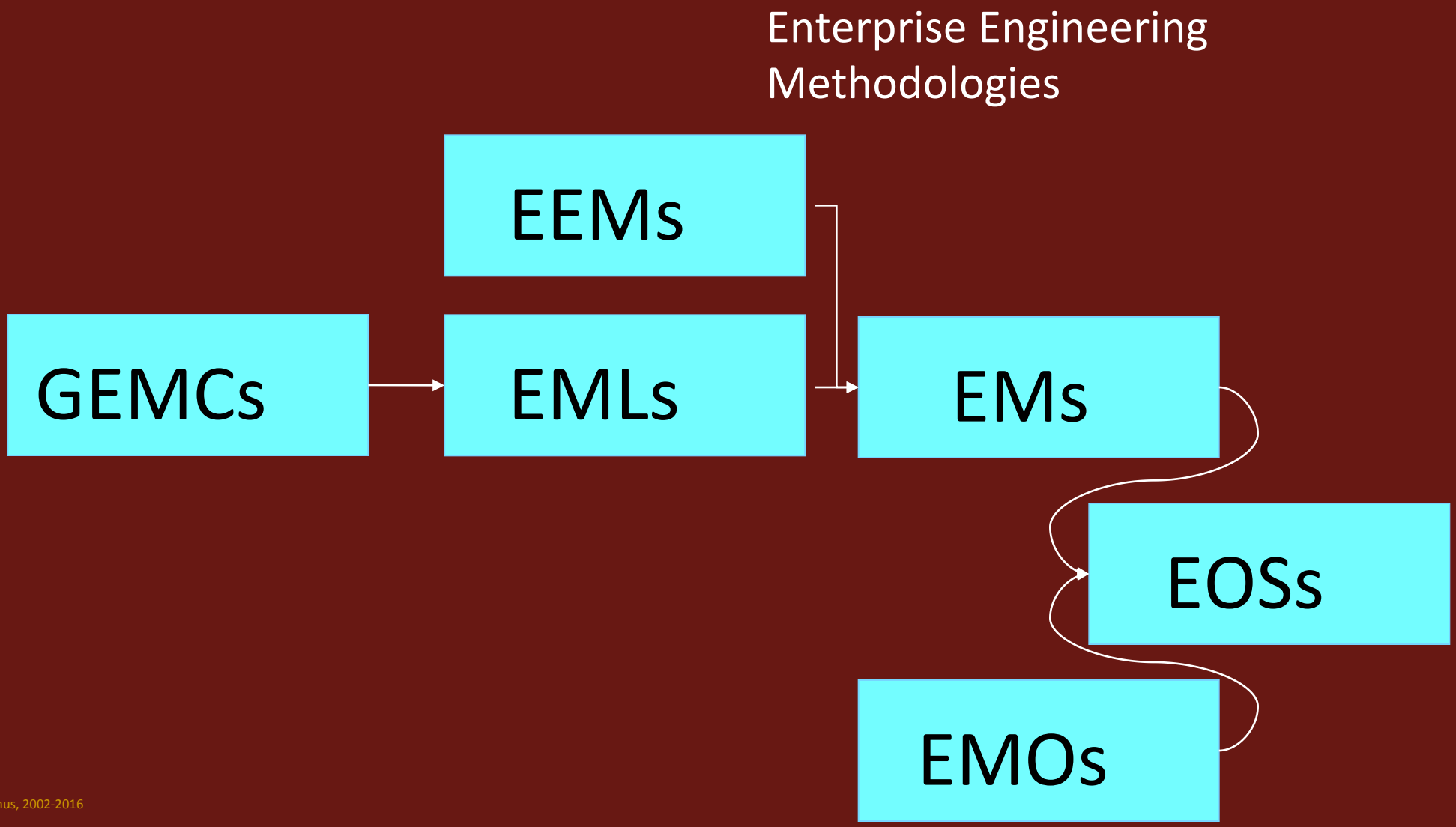


The Components of GERAM

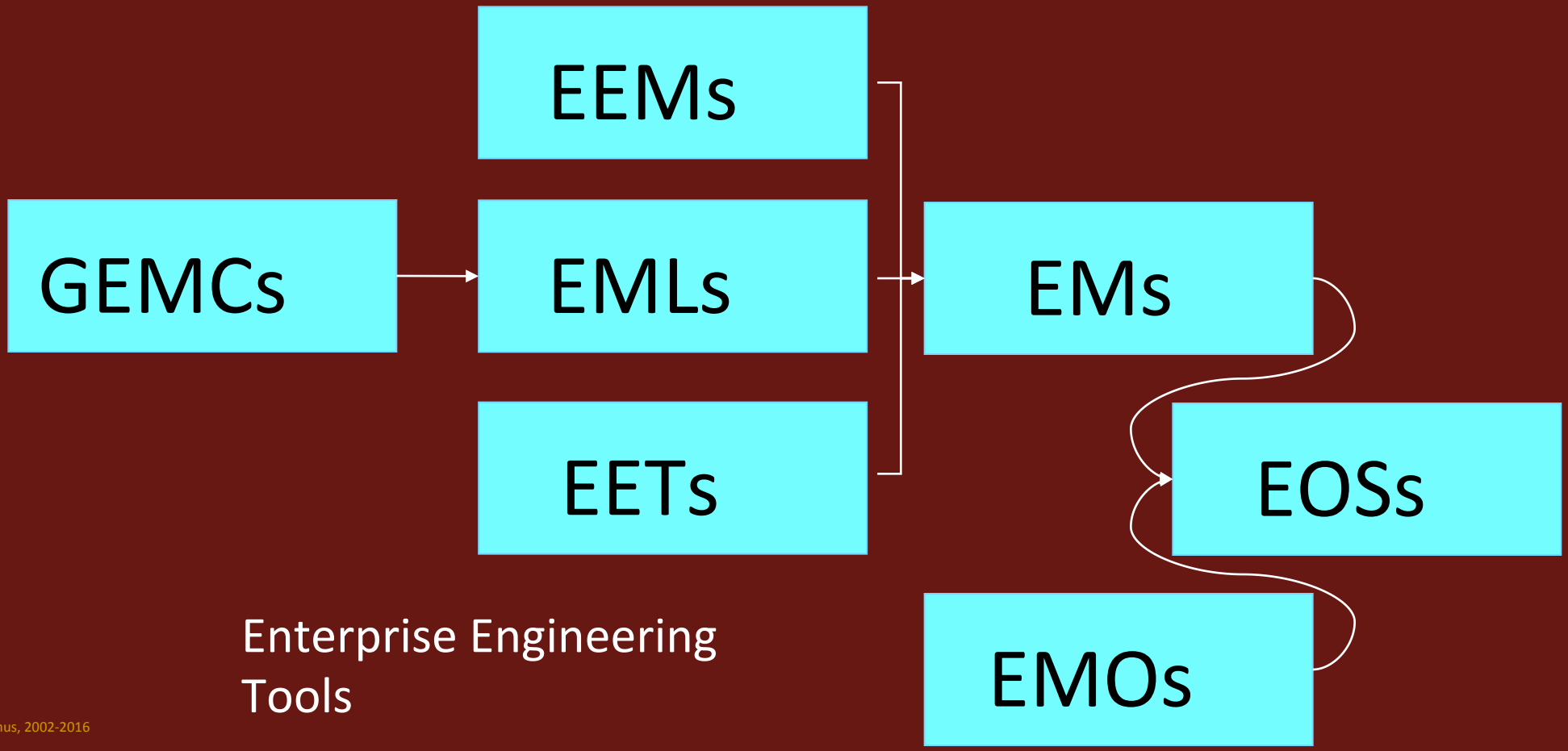
Generic Enterprise
Modelling Concepts



The Components of GERAM

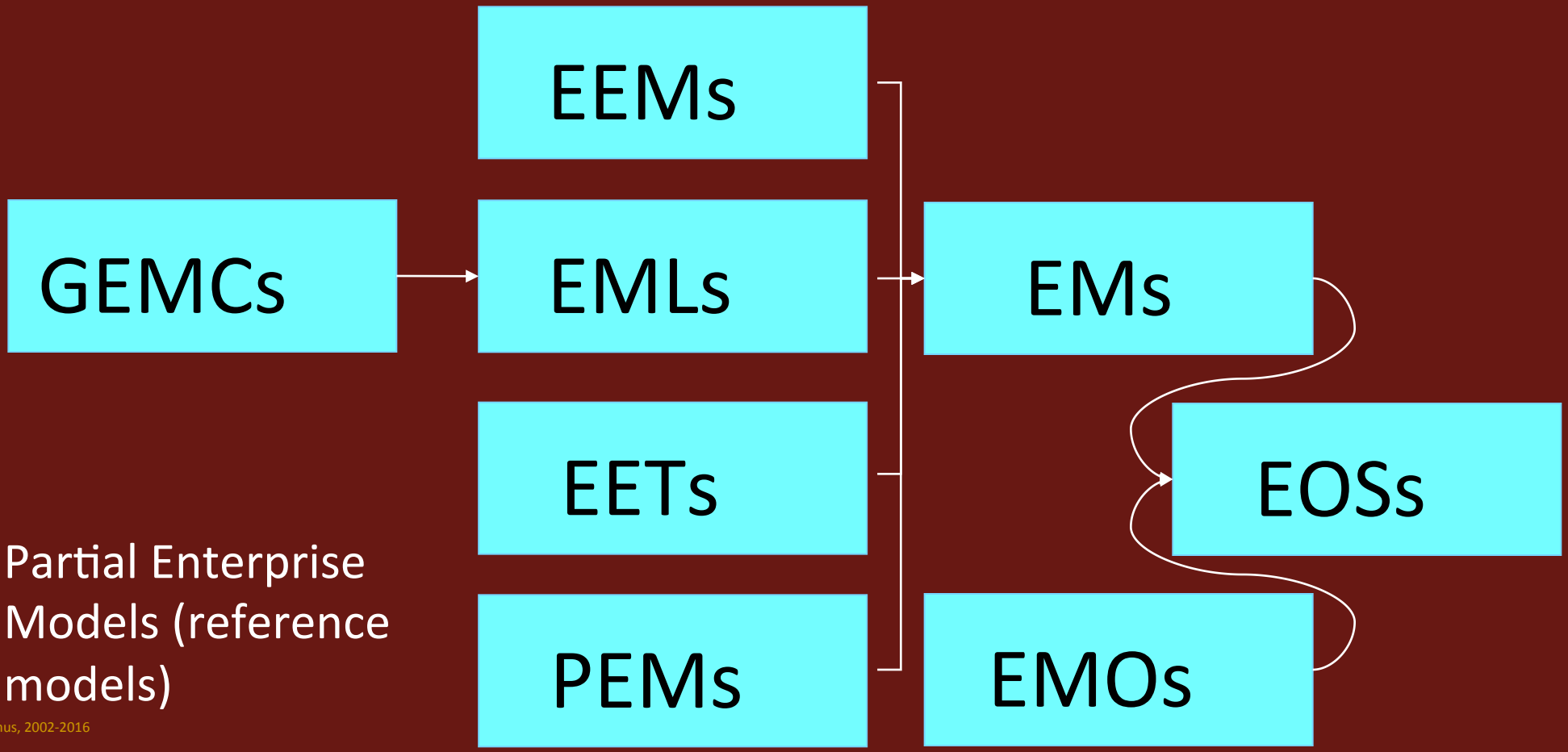


The Components of GERAM



Enterprise Engineering Tools

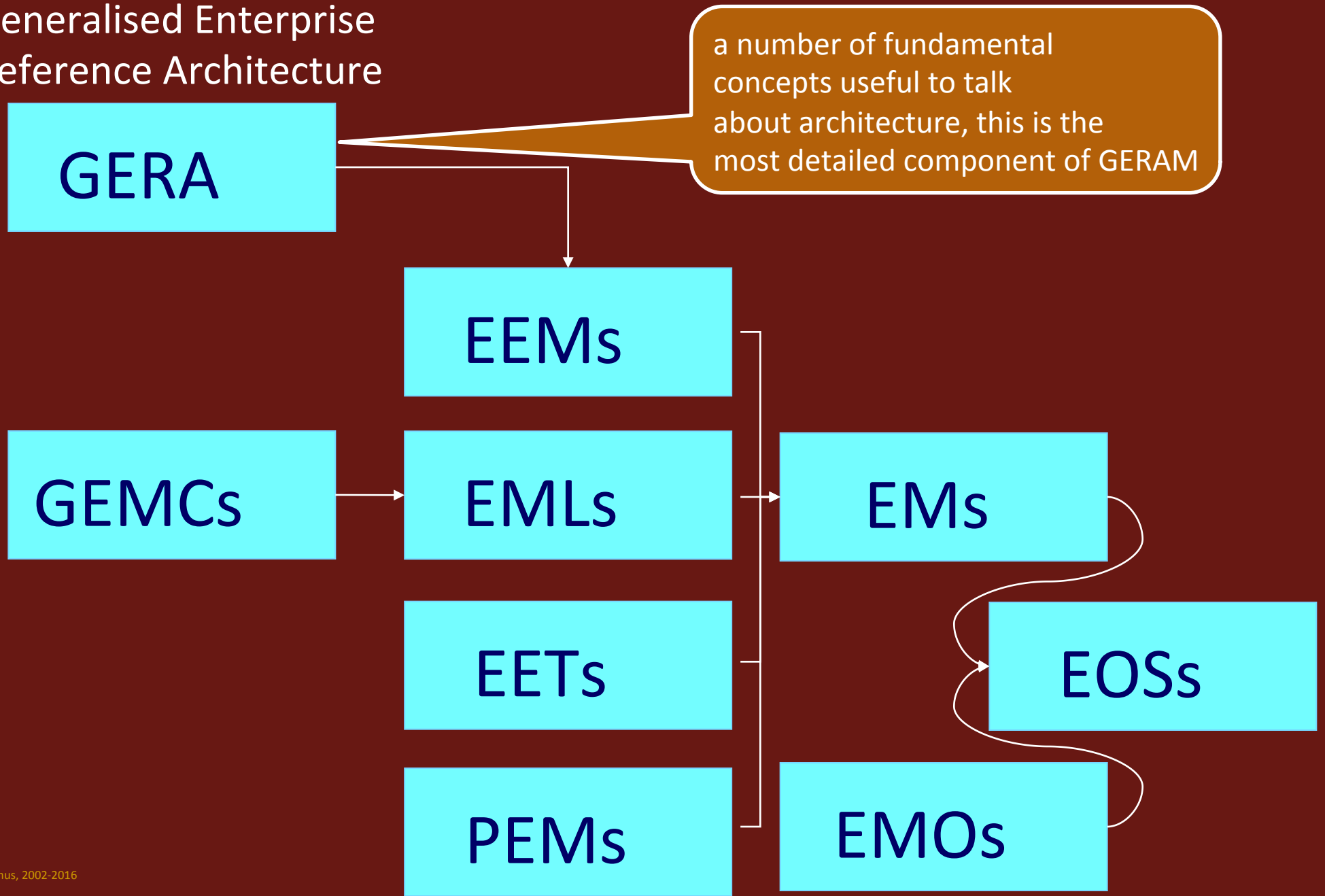
The Components of GERAM



Partial Enterprise Models (reference models)

The Components of GERAM

Generalised Enterprise Reference Architecture



GERA
(Generalised Enterprise Reference Architecture)

- The most detailed component of GERAM
- Defines fundamental concepts
  - Enterprise Entity types
  - Life-cycle
  - Entity recursion
  - Life history
- Defines an Enterprise Modelling Framework (GERA Modelling Framework)

Enterprise Entity types

- Repetitive (sustained)
  Service and/or Manufacturing entity

  (Note that 'service' may even include pure management services provided to another entity)

- *Project* enterprise

- *Product* entity

The above types are not an exhaustive list, users of GERA concepts could define their own entity types – the above are only typical types

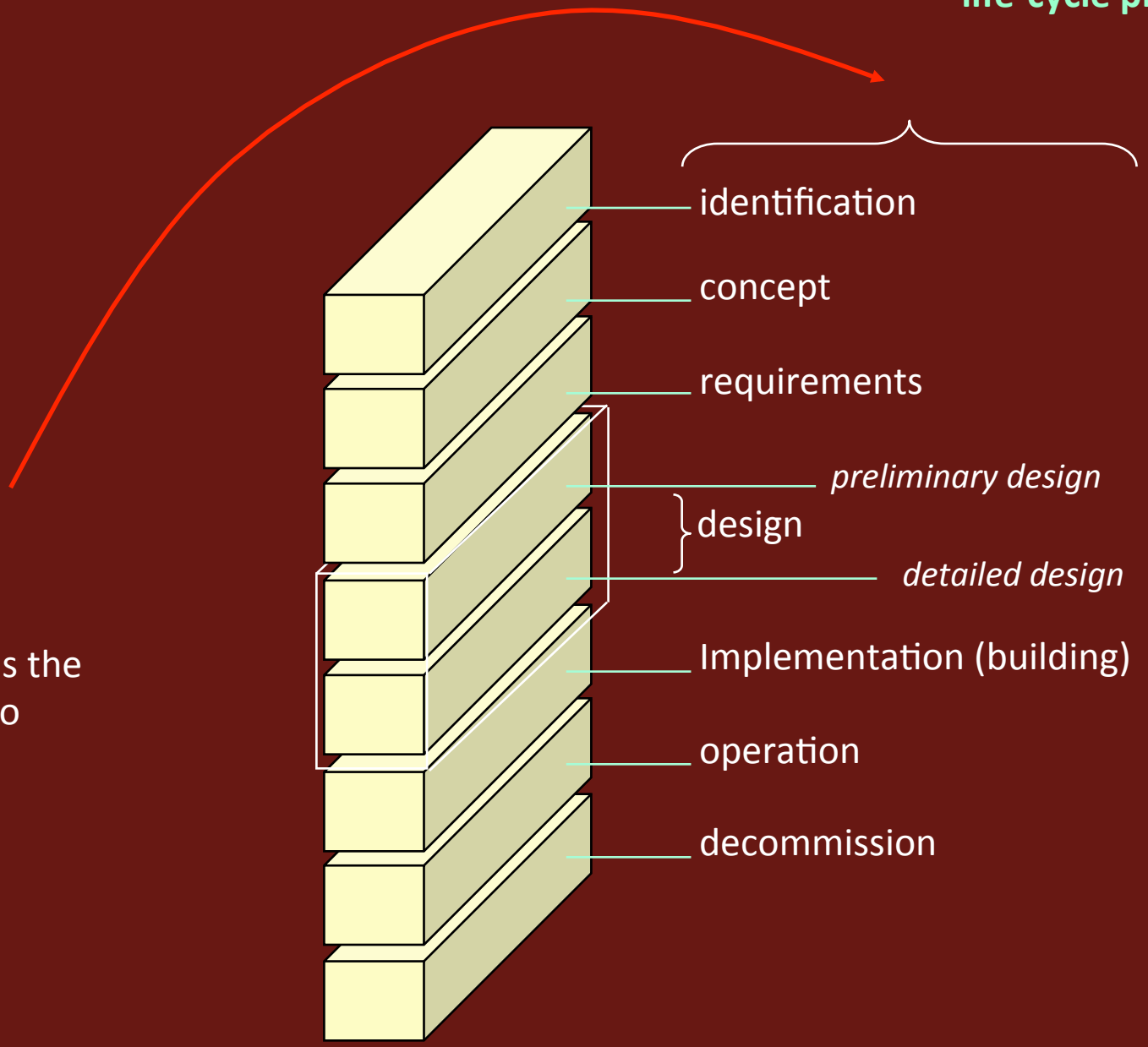
Life cycle (LC)

Each of these is called a 'life-cycle phase'

The life-cycle of an entity is an abstraction:

it lists the *types of activities* that can be done to and entity

… and consequently the we can deduce from this the competencies needed to cover the life cycle of the entity



(note: life cycle phase is *not* a temporal concept)

Entity recursion

- One entity may be involved in

  identifying,
  conceptualising,
  specifying,
  designing,
  building,
  supporting the operation, or
  decommissioning

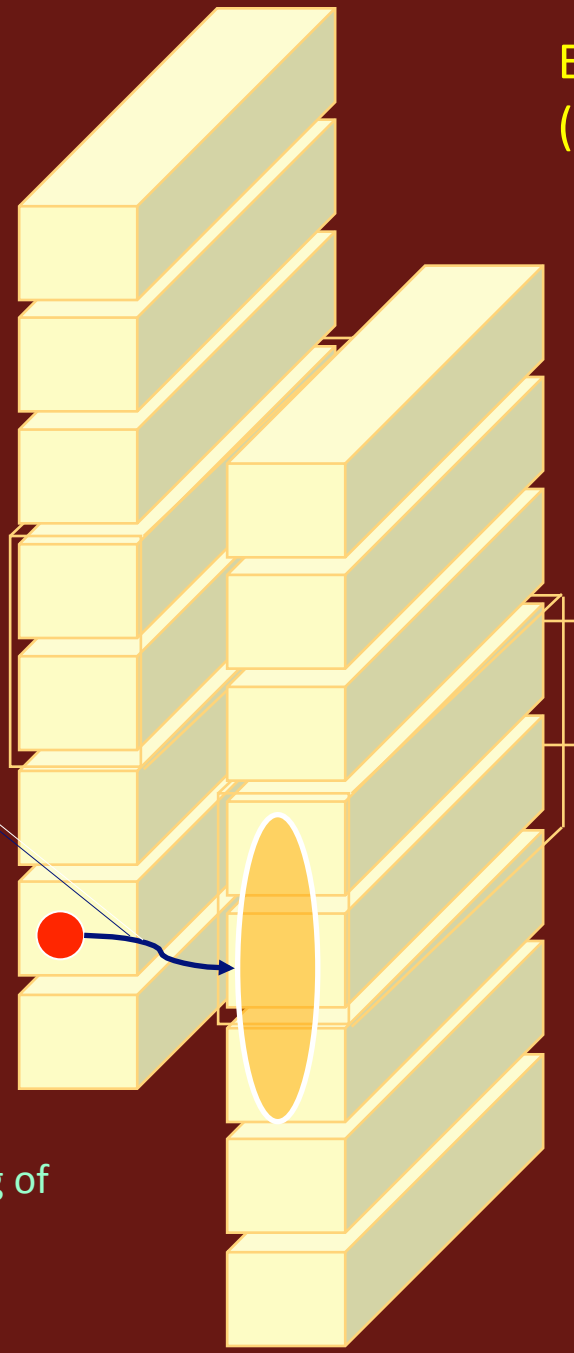
  of another entity

which in turn might do the same to other entities etc…

Entity A
(e.g. factory)

Entity B
(e.g. product)

Recursive relationships between life-cycles



identification

concept

requirements

preliminary design

design

detailed design

implementation

operation

decommission

This is called a generative relationship

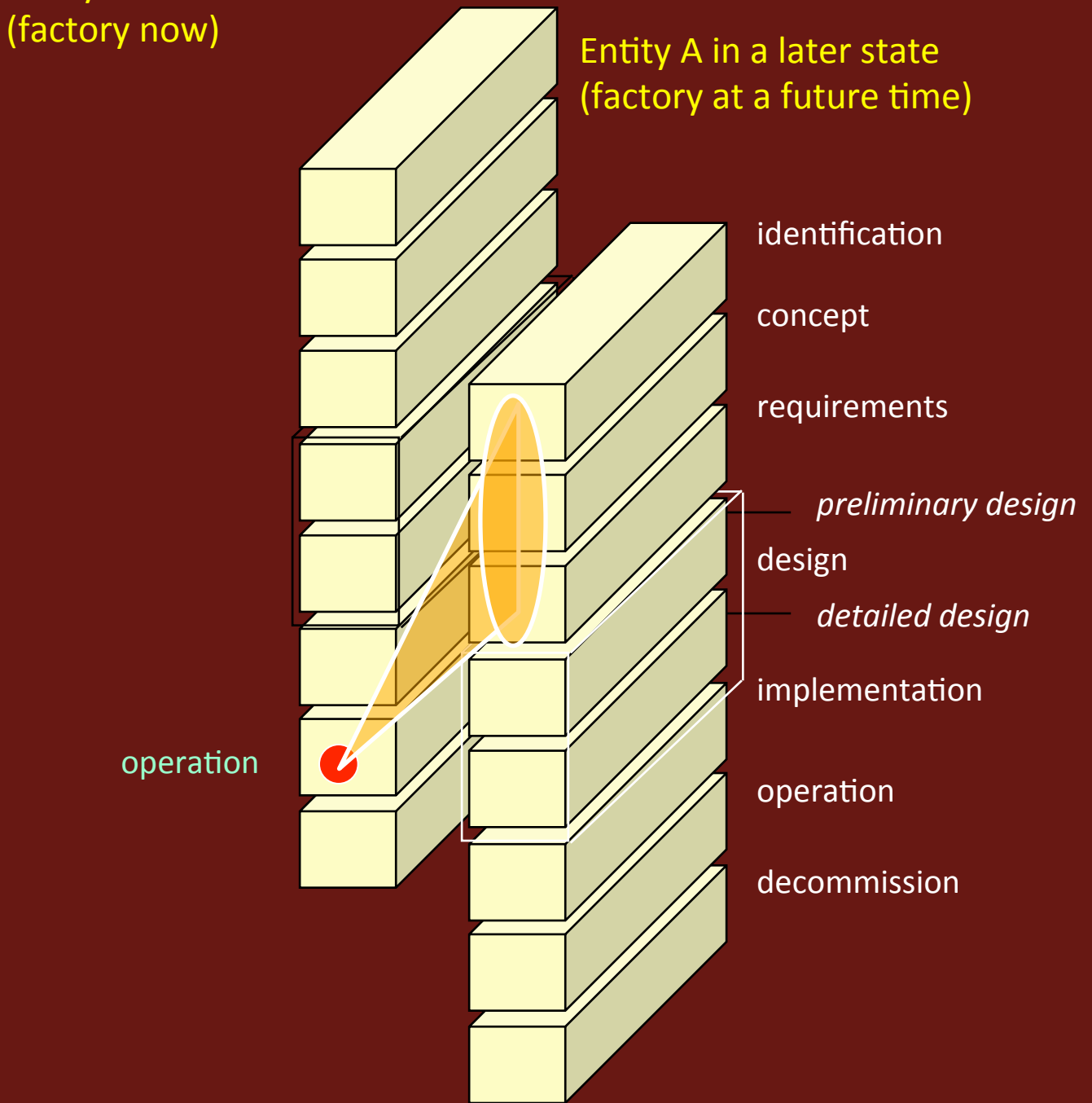
Entities are related through the operation of ‘A’ is supporting some LC activity(ies) of ‘B’

In this example ‘A’ (the factory) supports the preliminary design, detailed design and building of ‘B’ (the product).

Recursive relationships between life-cycles

As an entity operates it may perform some LC activity *on itself*

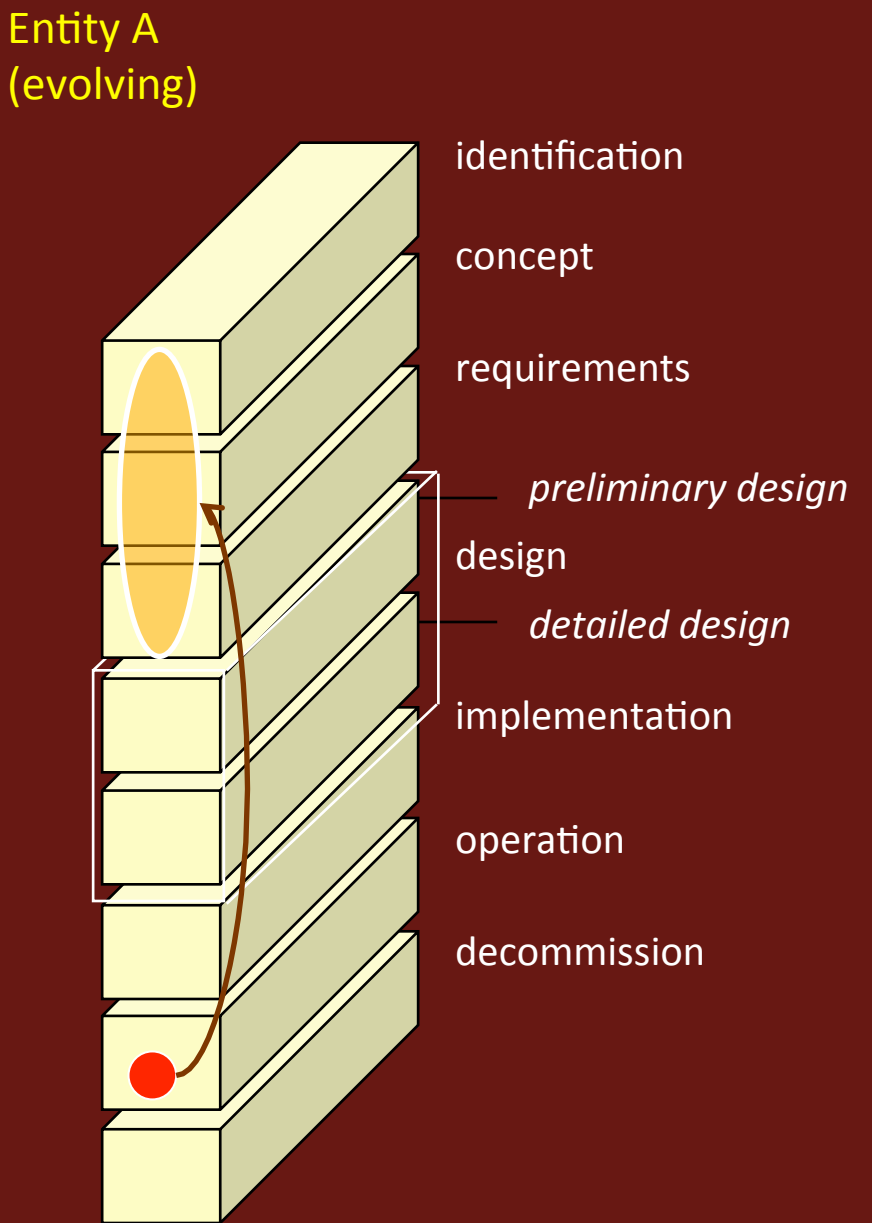
E.g. an entity could <u>design its own future state</u> (and control its own destiny!), re-build a part of itself, etc.



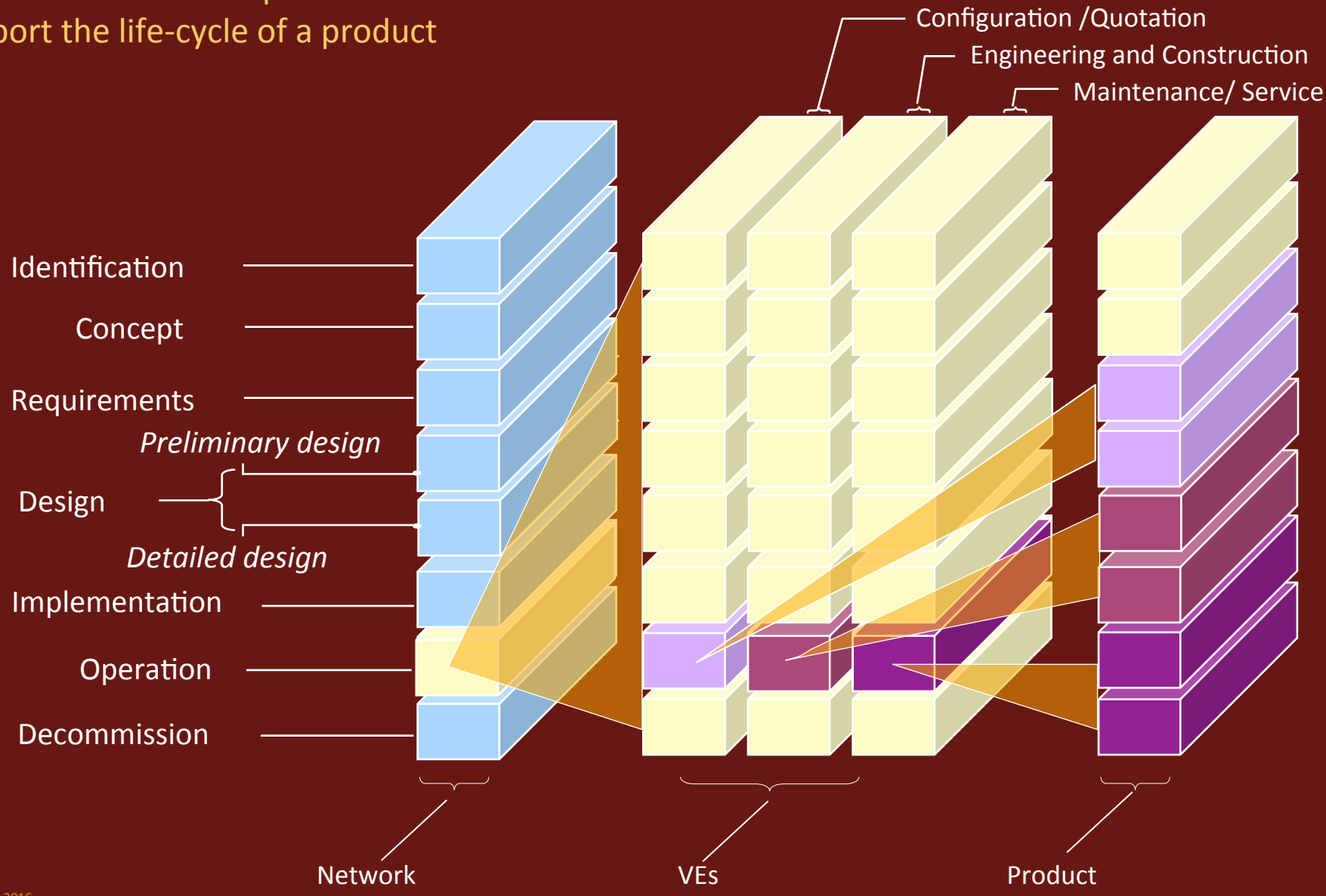
As an entity operates
it may perform
some LC activity
on itself

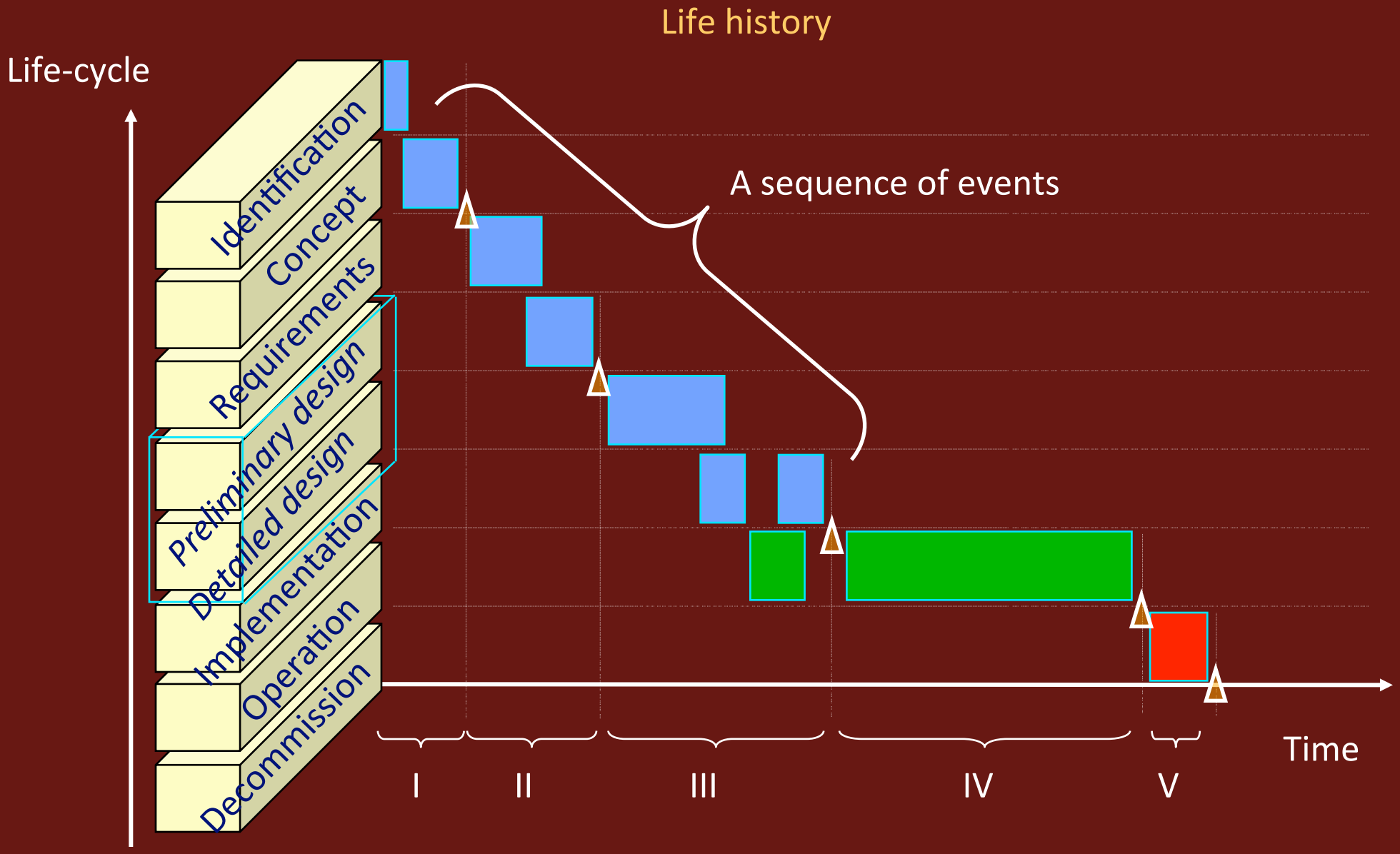
E.g. an entity could <u>design its own future state</u> (and control its own destiny!), re-build a part of itself, etc.

or to represent the same in a simpler way …



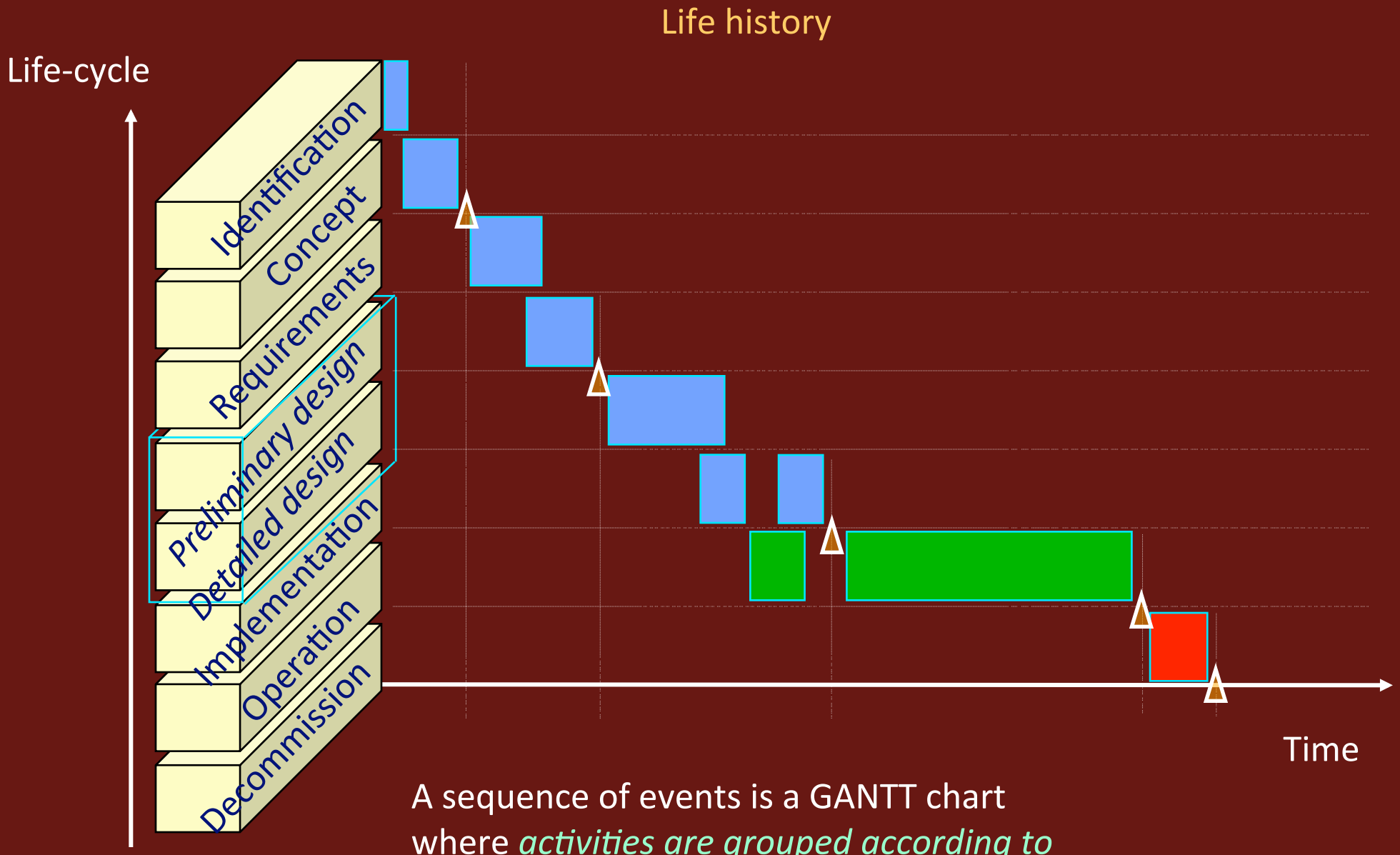
Example: a network of companies
may create ‘virtual enterprises’
to support the life-cycle of a product





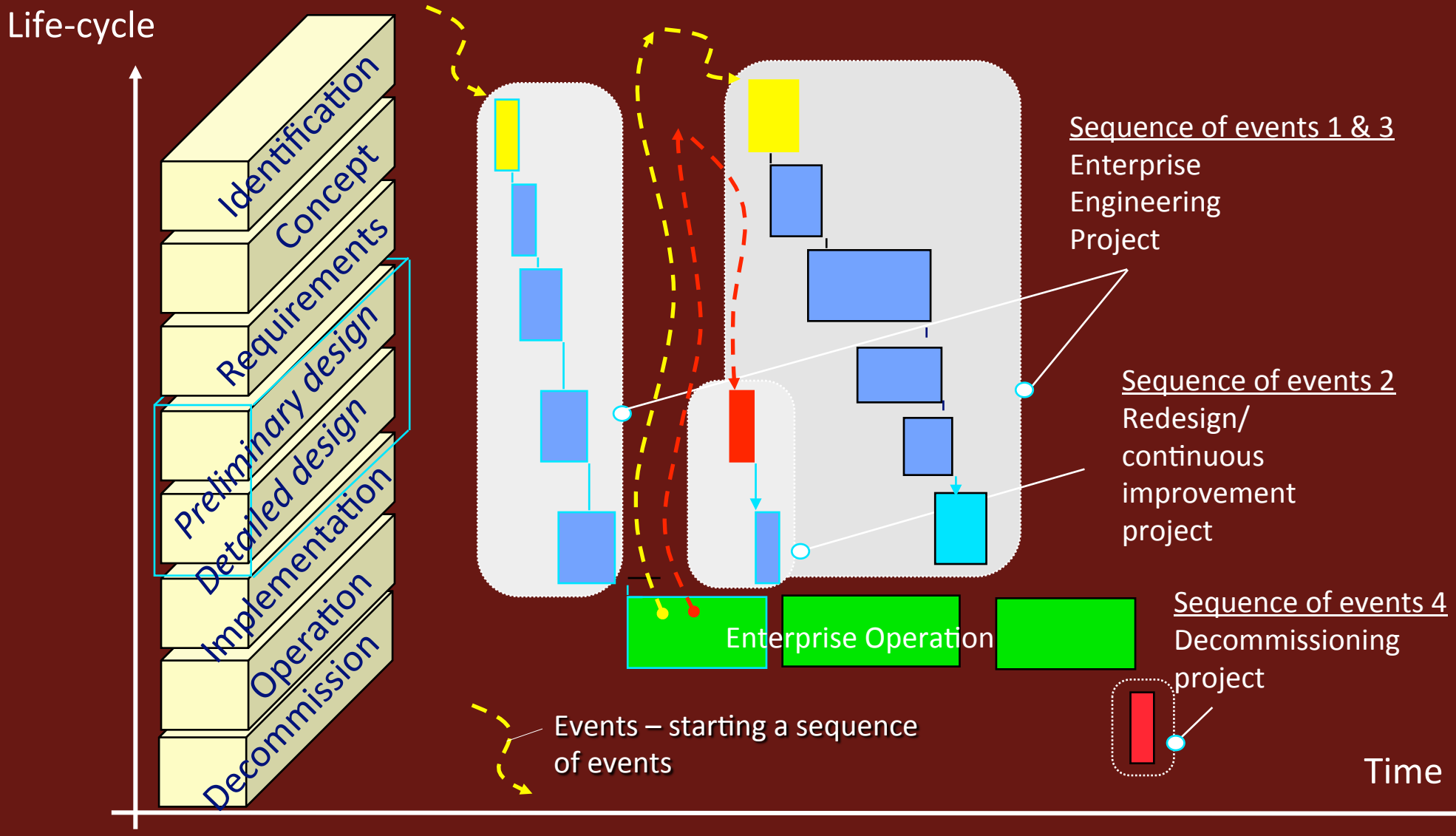
I, II, III … = Life history stages

△ = Milestones



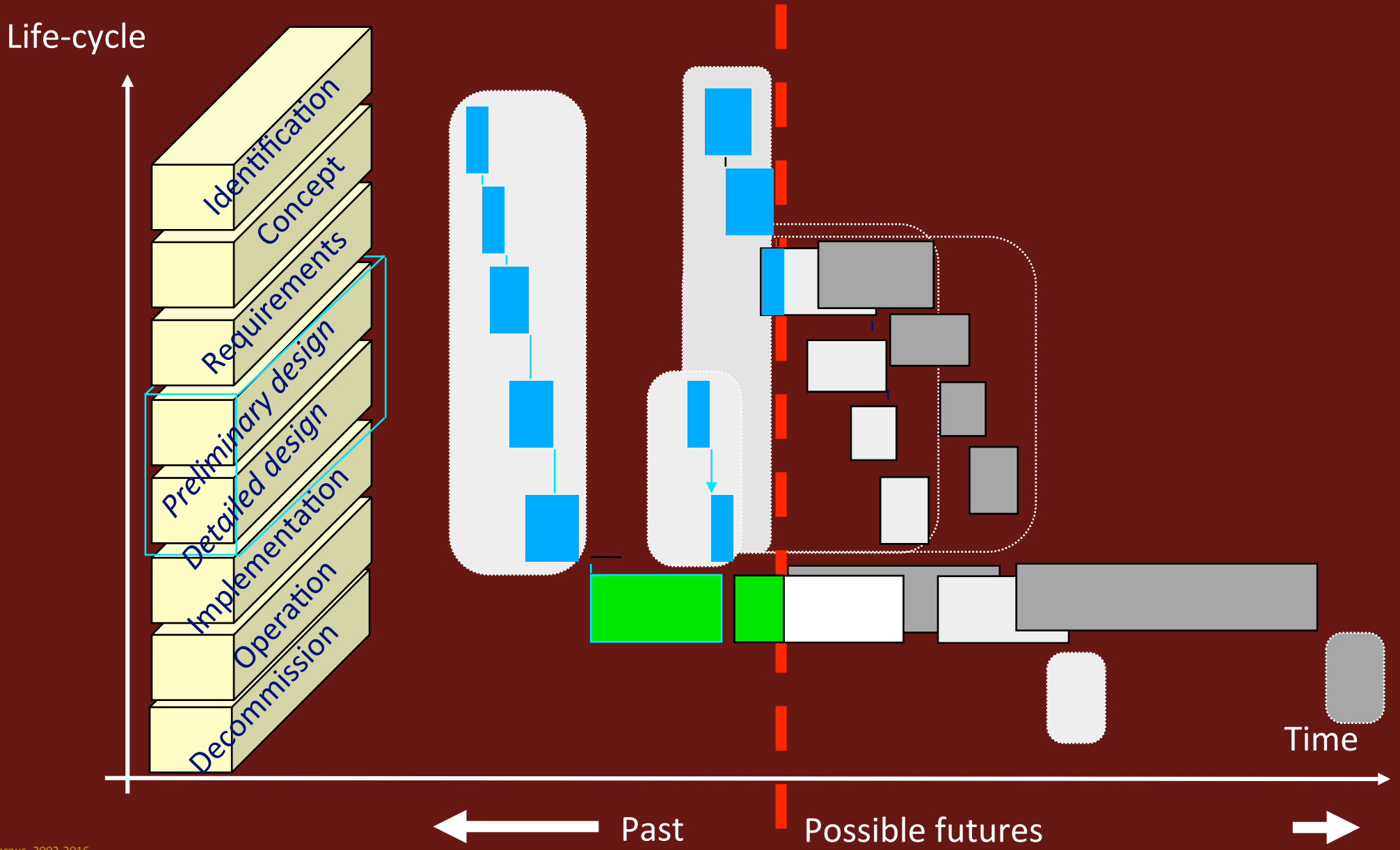
A sequence of events is a GANTT chart where *activities are grouped according to their life-cycle activity type* (‘phases’).

The life history of an entity consists of multiple, potentially parallel, sequences of events during the life of the entity

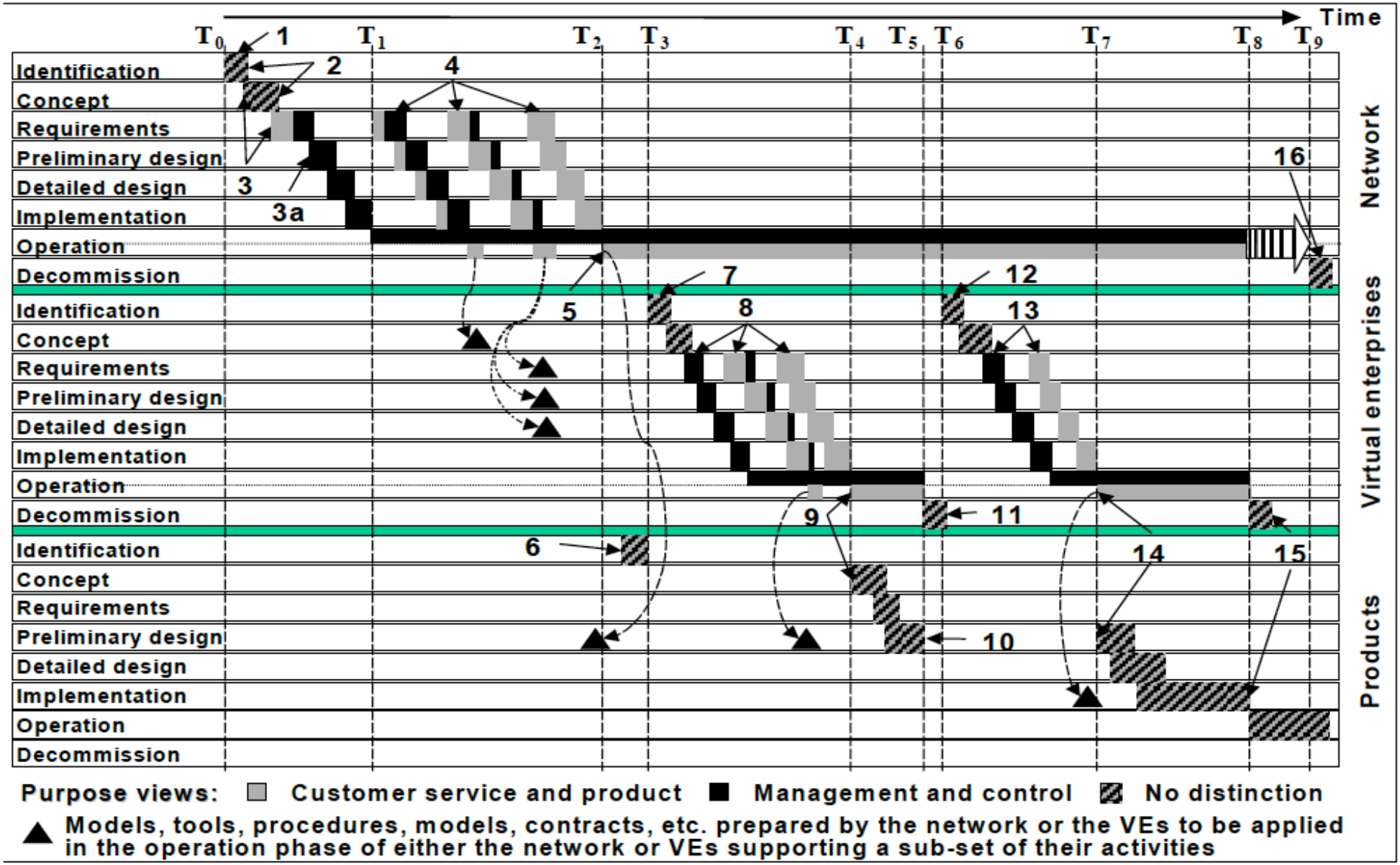


Life-cycle activities overlap (within / between sequences of events)

There is only one past history and there are multiple potential 'future histories' (futures)



Example: parallel life histories of three entities



Source: GMN Consortium

Summary

- Enterprise Architecture Frameworks organise knowledge necessary for EI
- None are complete
- ISO IS 15704:2000 & 2006 defines the requirements for them to be complete
- Introduced some notable frameworks
  (PERA, GRAI-GIM, CIMOSA, Zachman, ARIS, C4ISR/DoDAF, etc)
- Started the discussion of GERAM with GERA
  - enterprise entity
  - life-cycle
  - recursion
  - life history

Next

- GERA Enterprise Modelling Framework

The End