

CAD Software

A presentation providing
a comparative diagnosis of using these tools

By
Charles Meyer Richter RA0
(Ripose Architect grade zero)
Ripose Pty Limited

What is CAD?

Computer-aided design (CAD) is the use of computers (or workstations) to aid in the creation, modification, analysis, or optimization of a design.

Examples of CAD tools

The following are 6 software products used by any practitioner involved with designing 'information systems':

- Proprietary:
 - Alfabet
 - ArchiMate
 - PowerDesigner
 - Rational Rose
 - Sparx
- Free E/R tools

They all have the following in common:

- Technology-centric people
- Drawing a box to represent an 'entity' or a 'process'
- Drawing a line to connect boxes to represent an association or relationship
- Positioning boxes on a page

The problem

Every CAD tool was designed by software developers and focused on 'projects' (technology-centric) rather than on actual business-centric requirements.

The sections:

Before I get into the discussion I would like to clear up a few definitions. These are my understandings of the words I will use in this presentation. You may wish to find synonyms but basically they all provide a means of finding Common Ground.

Word	Class	A Definition
people	noun	Plural of person "a human being"
project	noun	"A planned undertaking"
entity	noun	"a thing with distinct and independent existence"
process	noun	"a series of actions or steps taken in order to achieve a particular end"
relationship	noun	"the way in which two or more people or things are connected, or the state of being connected"

The 5 sections

What are the 5 sections of CAD tools trying to represent?

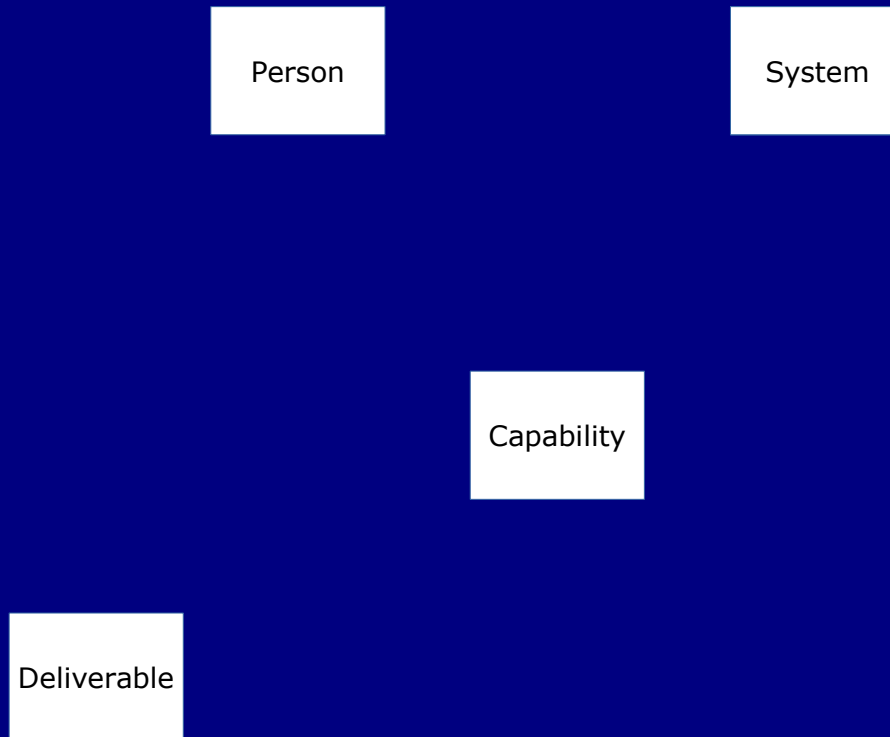
Word	Synonyms	
People	People	
Project	Systems	Deliverables
Entity	People	Deliverables
Process	Systems	
Relationship	Deliverables	Capabilities

The additional sections

The definitions of the representations:

Word	Class	A Definition
person	noun	"A human being"
system	noun	"Instrumentality that combines interrelated interacting artifacts designed to work as a coherent entity"
instrumentality	noun	"serving as a crucial means, agent, or tool"
artifact	noun	"A man-made object taken as a whole"
coherent	noun	"Capable of thinking and expressing yourself in a clear and consistent manner"
entity	noun	"That which is perceived or known or inferred to have its own distinct existence (living or nonliving)"
capability	noun	"The susceptibility of something to a particular treatment"
deliverable	noun	"a thing able to be provided, especially as a product of a development process"

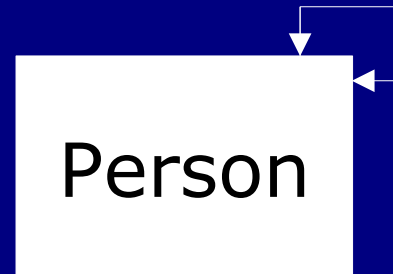
The four 'things' that need to be linked graphically are:



It is not that easy as there are a number of interfaces that need to be considered:

- People and People
- Systems and Systems
- People and Systems
- People and Capabilities
- Systems and Capabilities
- People, Systems and Capabilities
- Deliverables and Deliverables
- People, Deliverables and Capabilities
- Systems, Deliverables and Capabilities

A Person can interface with a multitude of People. This can be represented by the following diagram:

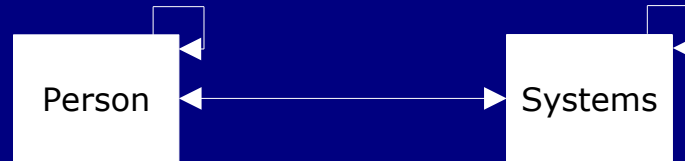


To simplify this I will use the following notation:

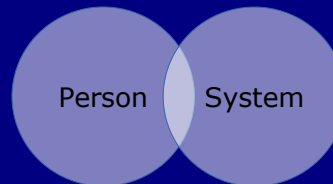


A Person can interface with a multitude of Systems and a System can effect a multitude of People. This can be represented by the following 2 diagrams:

A simple relational view:



Or as a Venn diagram:

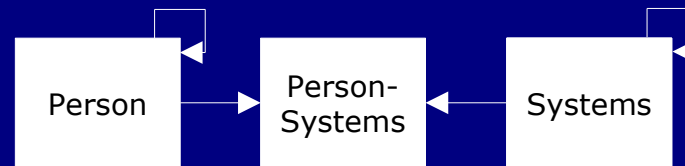


As Venn diagrams can become messy I will dispense with them.

The Person to System Interface

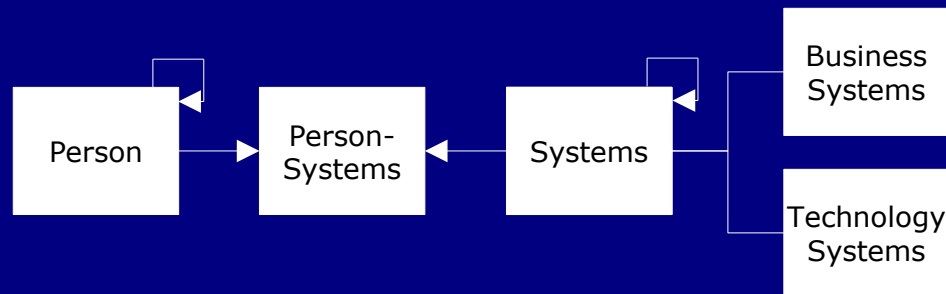


This representation of a “many to many” relationship can be simplified by placing an intersecting ‘thing’ between the the two as shown in the following diagram:



Later on these ‘things’ will become known as ‘entities’

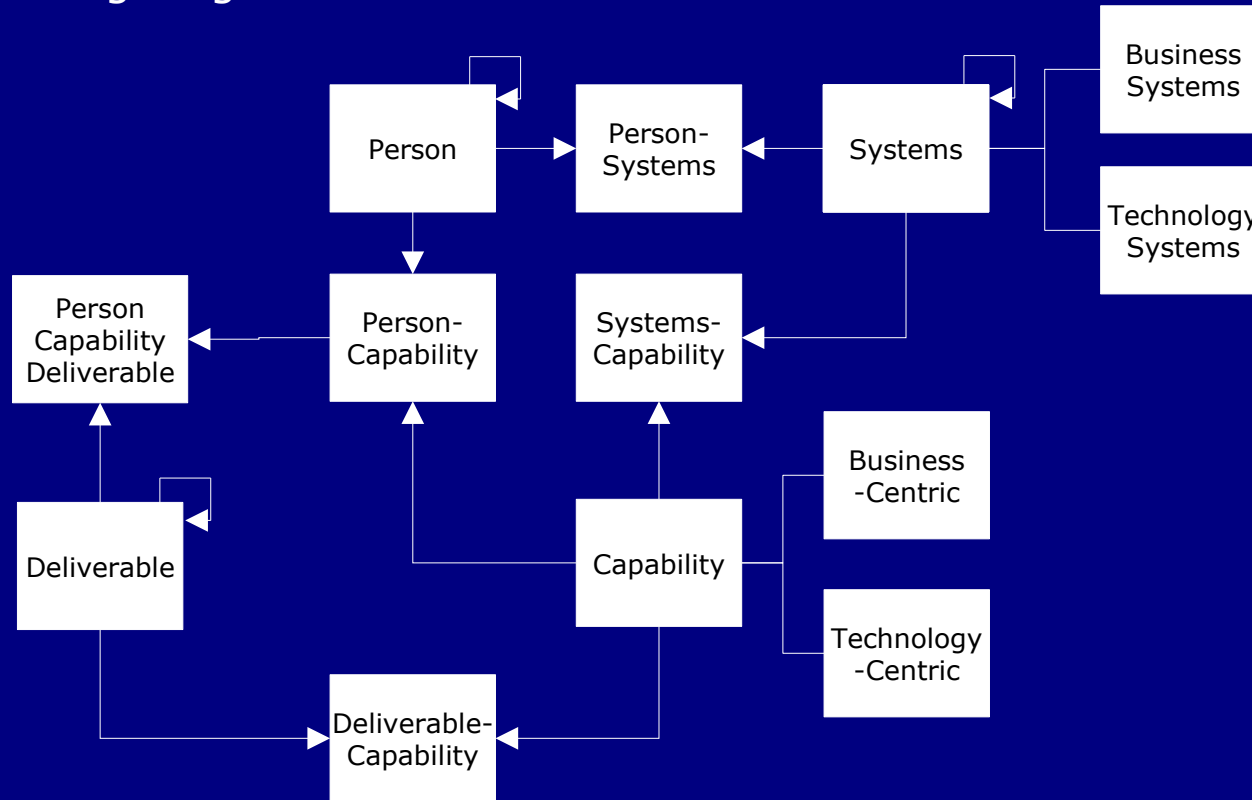
Now let us consider how a system can be best described. There are only two types of systems and these are shown in the following diagram:



Therefore by inference a 'Person' can now interface with 'Business Systems' and/or with 'Technology Systems'. This then removes the need to describe a 'Person' as being either 'business-centric' or 'technology-centric'.

Note: The relationship between 'System' and their sub-classes. This model describes that a 'System' can be either a 'Business System' or a 'Technology System' but not both. It also describes that either of these has to be a type of 'System'

Without me going into the logic as to how the rest of the components in the rest of the model can be individually constructed I will now present you with a composite view of all the interfaces as shown in the following diagram:

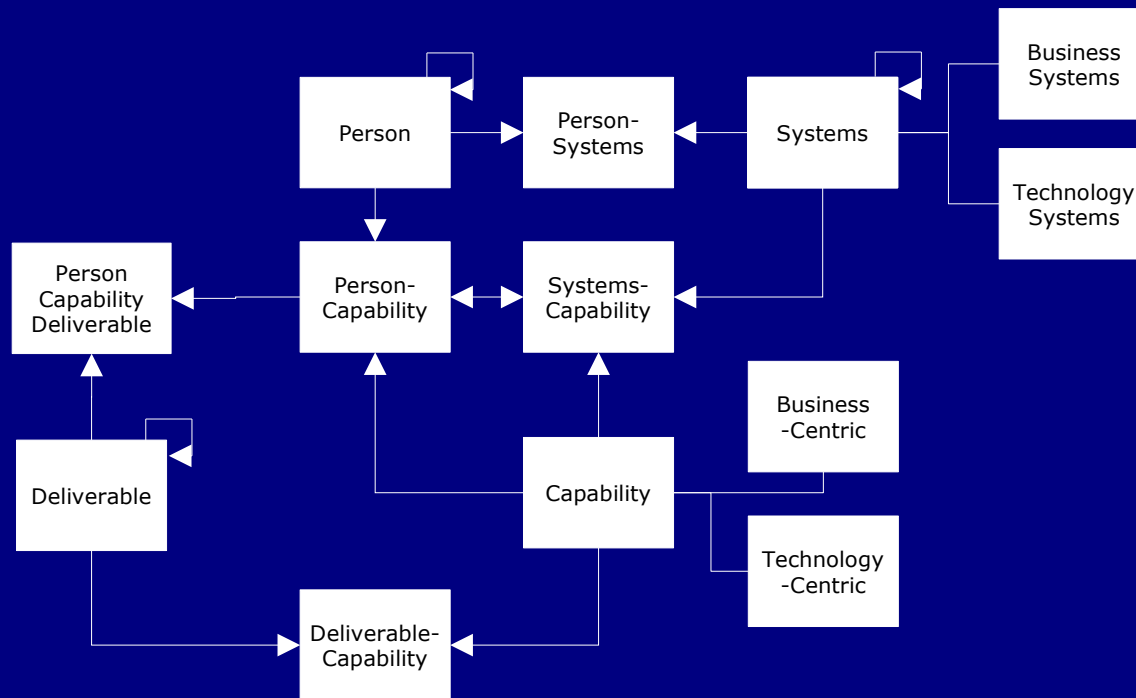


The missing Interface

If you look carefully at the previous model you may have noticed that there are a number of missing interfaces.

This is shown as a 'many to many' relationship going on between a 'Person Capability' and a 'Systems Capability'. What about the link between 'Systems Capability' and 'Deliverable Capability'? To put this into the diagram will only create more clutter.

These will need to be resolved as somewhere along the line someone will have to remove any conflict between the 'Capabilities' and this is where every approach, using this model, will begin to falter and ultimately end up as a 'systemic failure'.



A better solution

To remove the likelihood of wasting a great deal of resources (time and money) I have a better solution. One based on creating a different 'Universe of Discourse' which will prevent any 'systemic failure'.

I will begin by changing one 'thing'. Instead of the word 'System' I will introduce the notion of the 'Information System' and will start off by using following Wikipedia description which is:

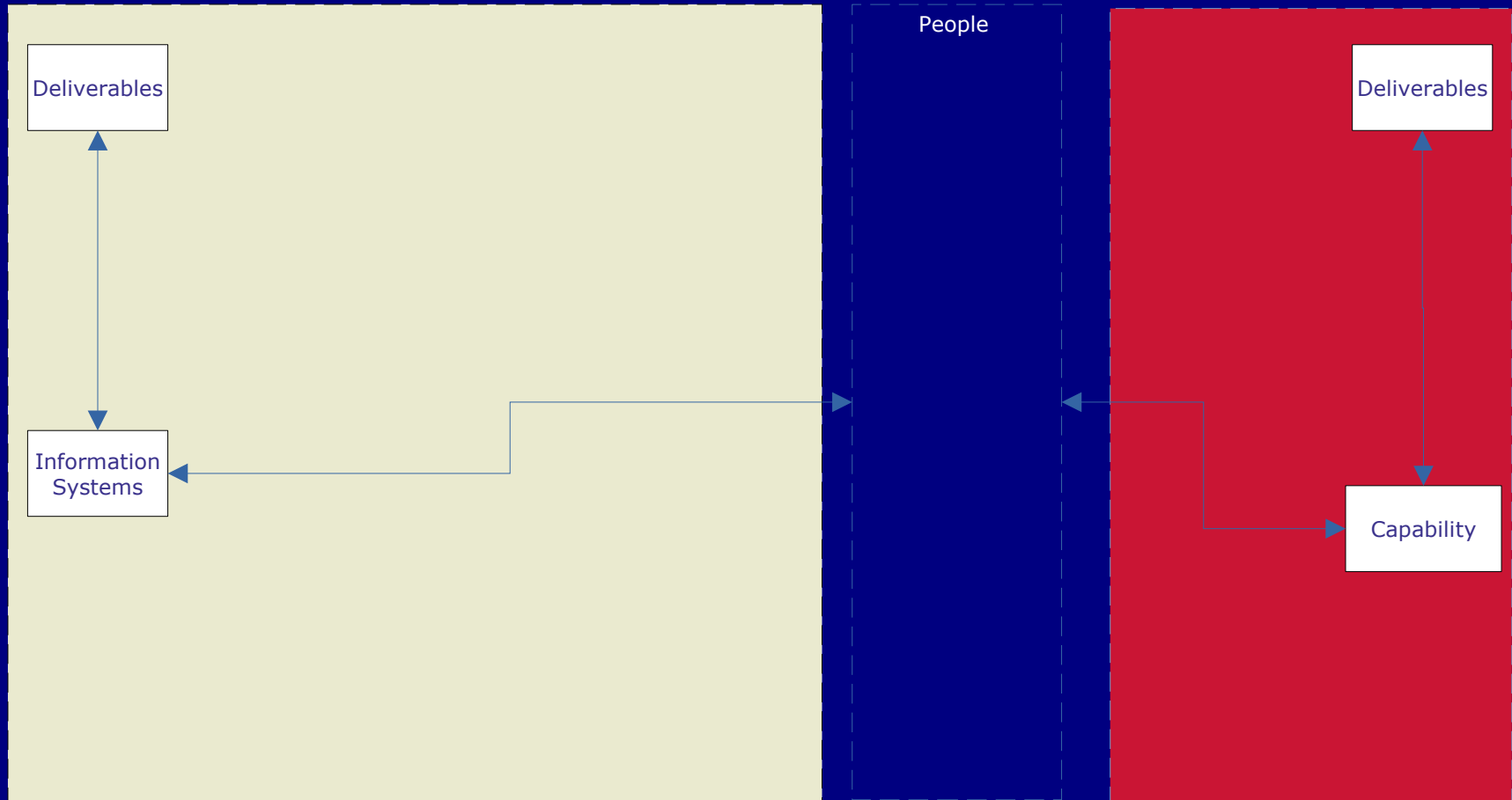
(noun) "A formal, sociotechnical, organizational system designed to collect, process, store, and distribute information. In a sociotechnical perspective, information systems are composed by four components: task, people, structure, and technology"

However I have to disagree with the collaborating people of this entry as I have an issue with their four level Hierarchy system of 'systems'.

The next slide will present the beginnings of my viewpoint which, if followed, will eliminate any likelihood of any 'systemic failure' by removing any possibility of missing a 'many to many' relationship between any of the components.

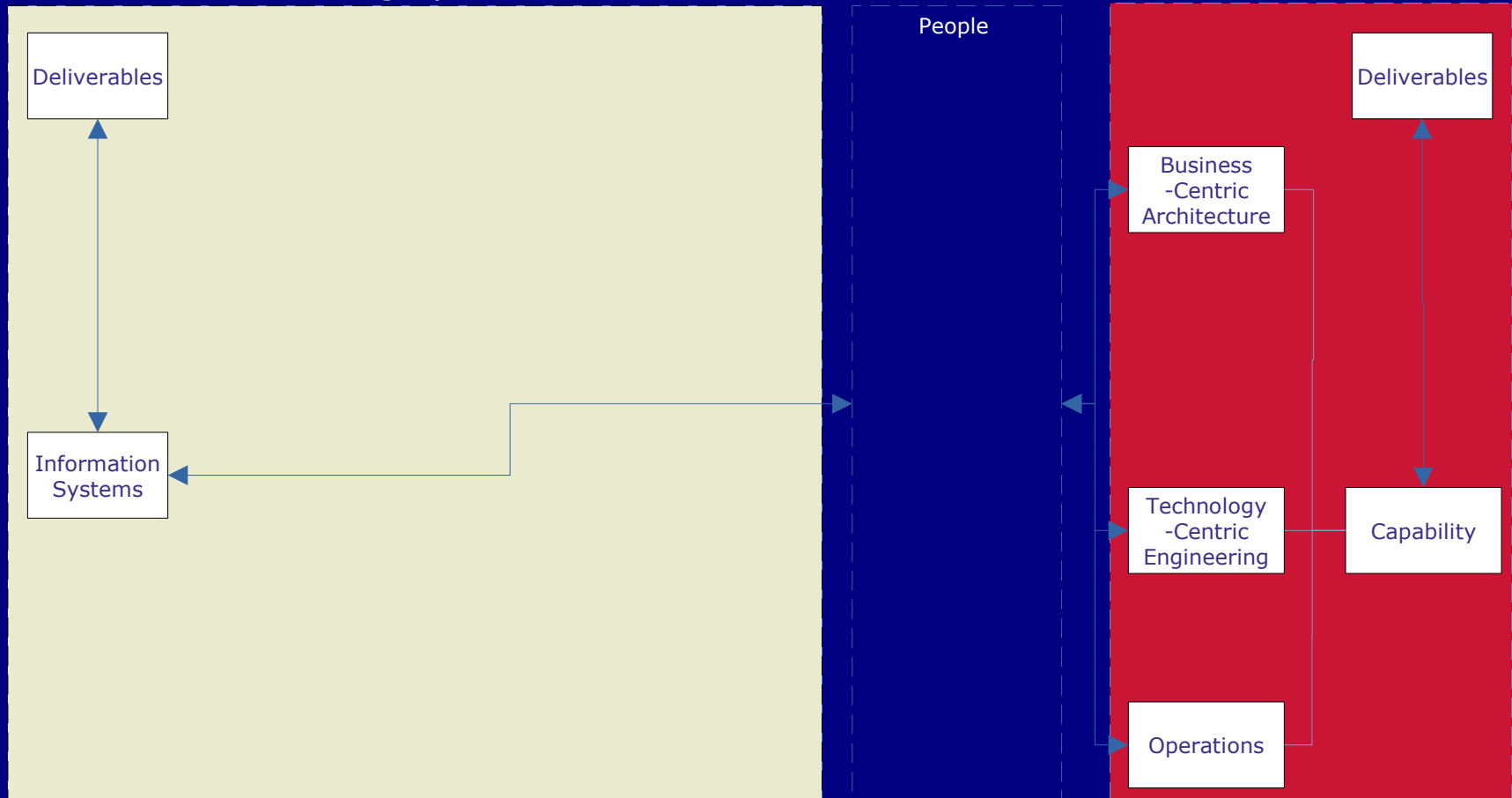
A better solution

To remove the likelihood of wasting a great deal of resources (time and money) I have a better solution. One based on creating a different Universe of Discourse which will prevent any 'systemic failure'. The following graphics represents how I view the 4 entities as described in slide 4. Note the change to the name of the 'Systems' component.



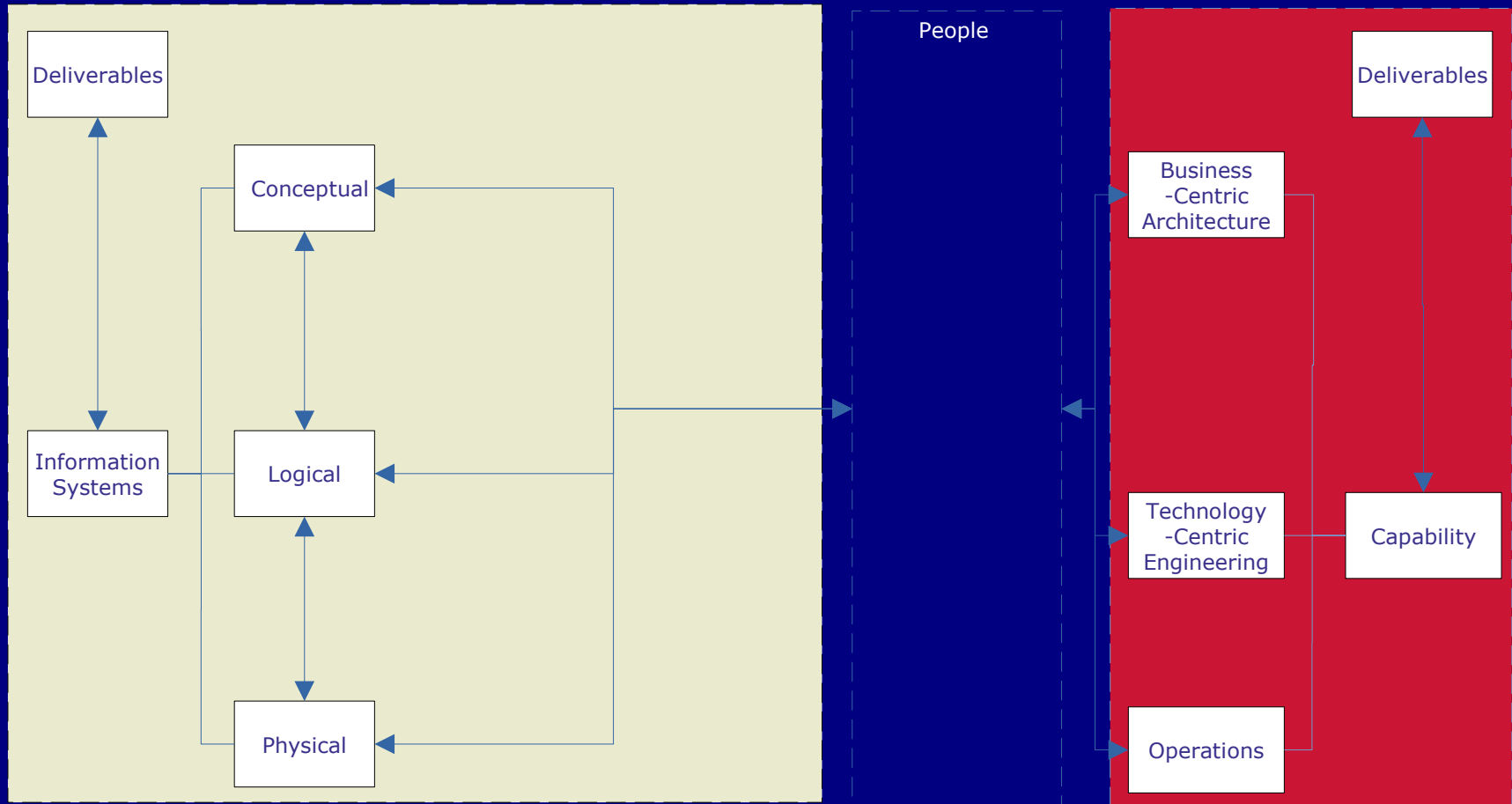
A better solution

1. 'Capability': I will resolve the 'many to many' relationship between a 'Capability' and a 'Person' by sub-dividing a 'Capability' into 3 sub types as shown in the below graphics.



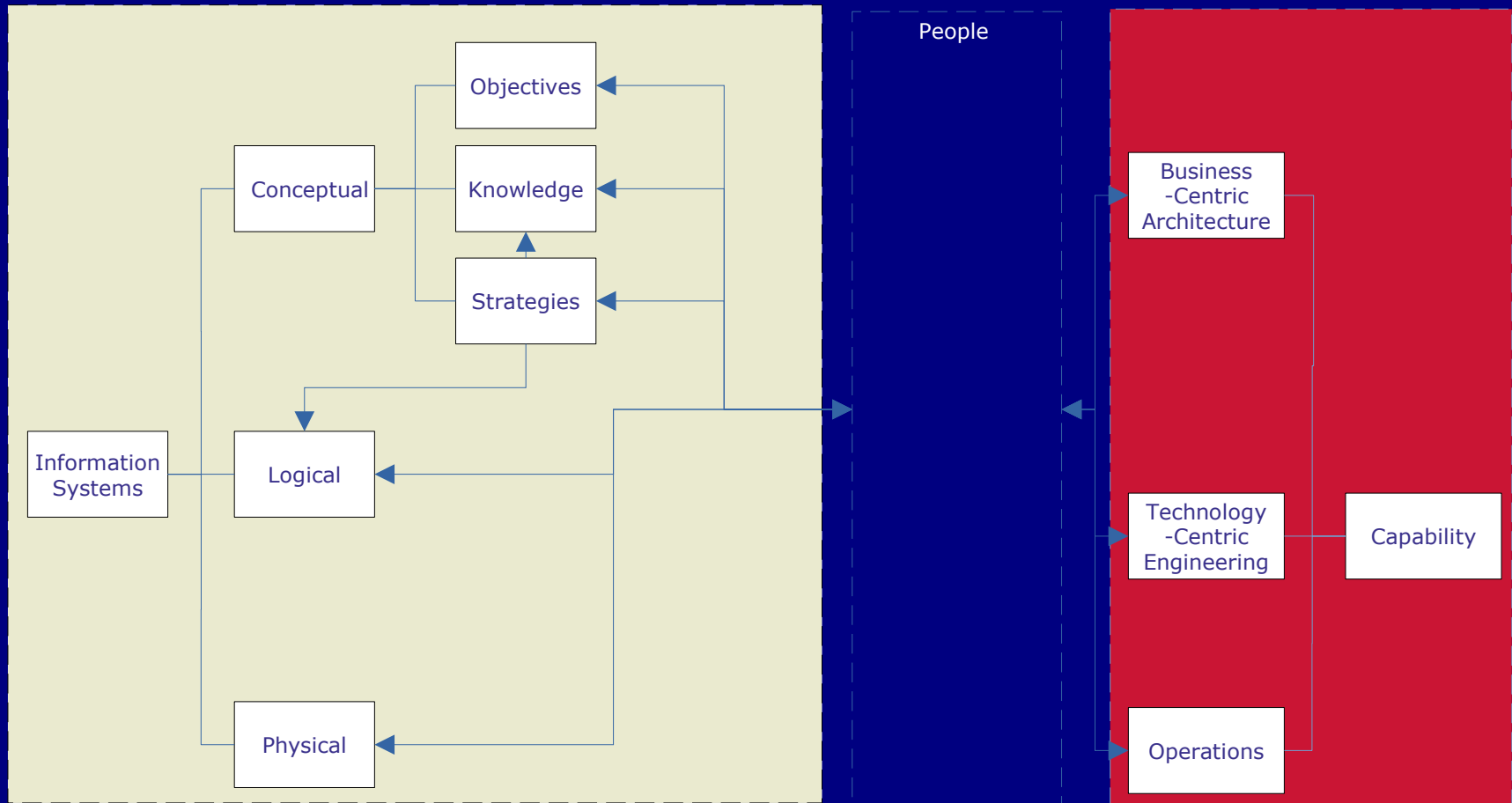
This introduces the notion that a 'Capability' is of the nature of either Architecture, Engineering or Operational.

2. 'Information Systems': I will now resolve the 'many to many' relationship shown on slides 7 through 11 by sub-dividing these into the 3 as depicted below. This will further reduce any confusions as to how one system relates to another.

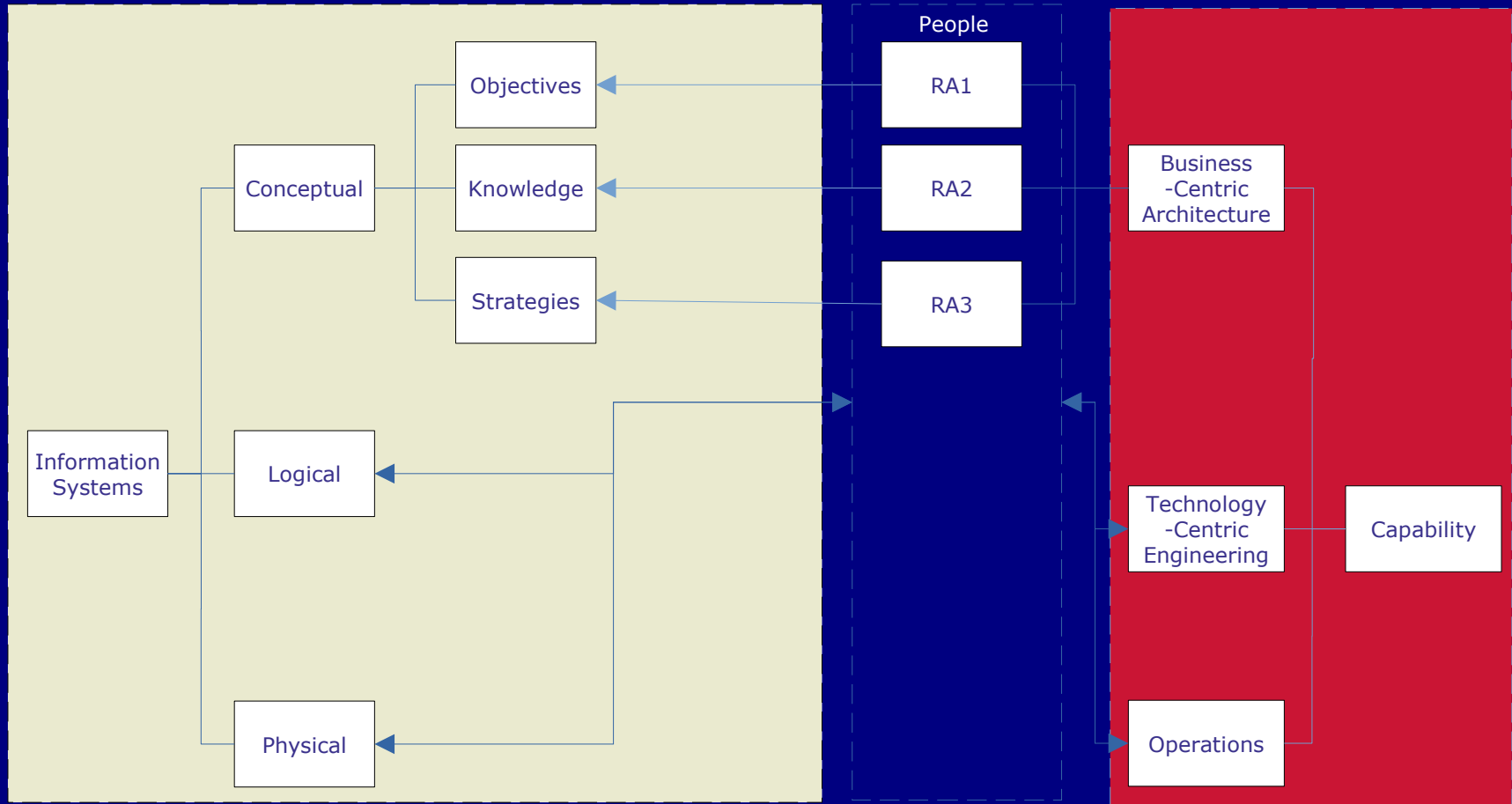


A better solution

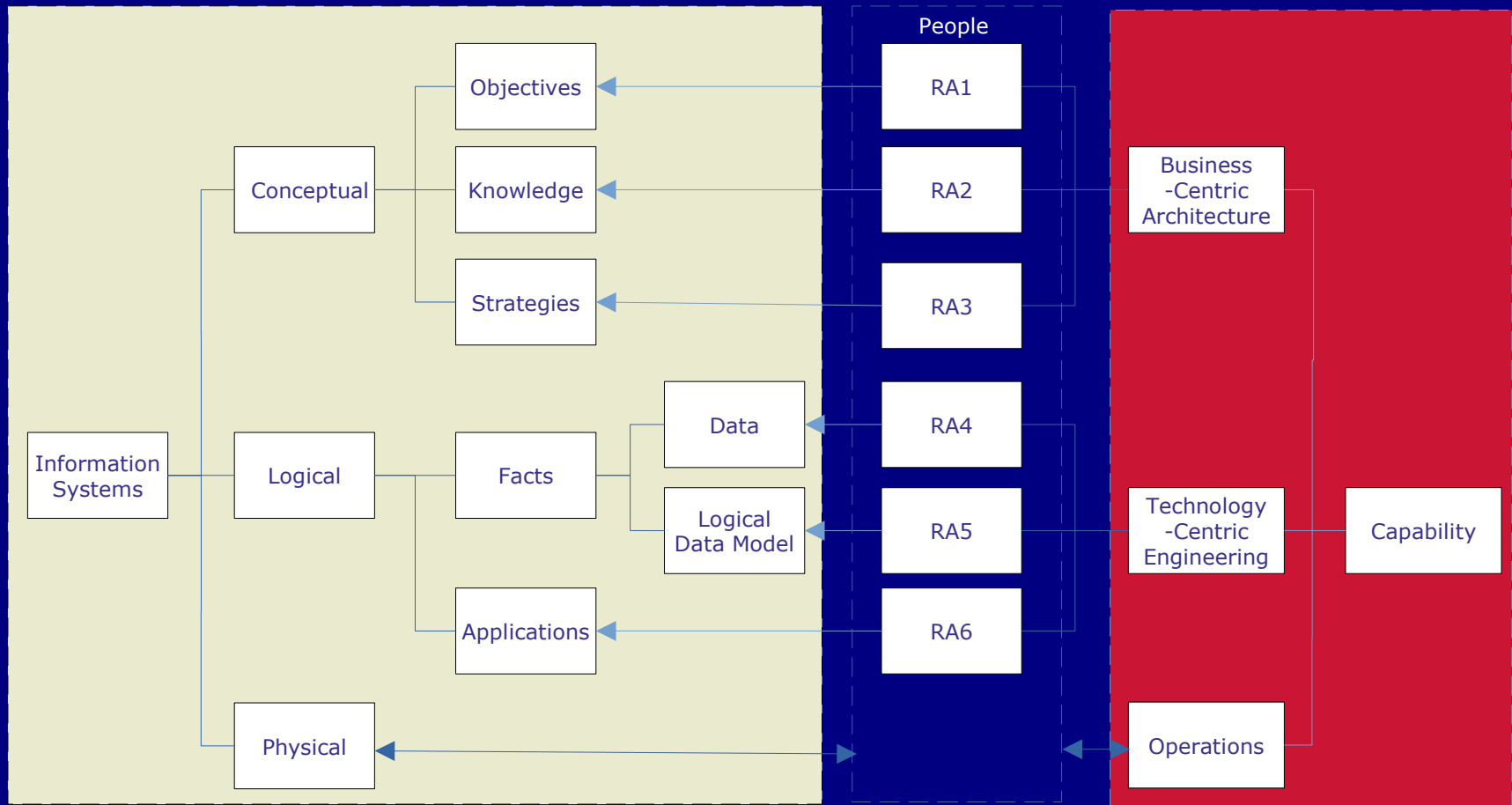
1.1. 'Conceptual': I will now sub-divide this domain into 3. In just about every approach the 'Knowledge' component has been ignored or replaced by either 'Data', the 'Conceptual Data Model' or the BABoK. All of which are Complex, Risky and Problematic. None of them seem to resolve the link between 'Knowledge' and 'Strategies' let alone between 'Objectives' & 'Strategies'. You may have noticed that the 'Deliverable' enigma has also began to be resolved by the introduction of the 3 'Conceptual Deliverables'.



1.1. The Conceptual Ripose Architects: I will now resolve the 'many to many' relationship between 'Conceptual Systems' and 'People' and between 'Architecture Capability' and 'People' by introducing the Ripose Architect grades 1, 2 & 3.

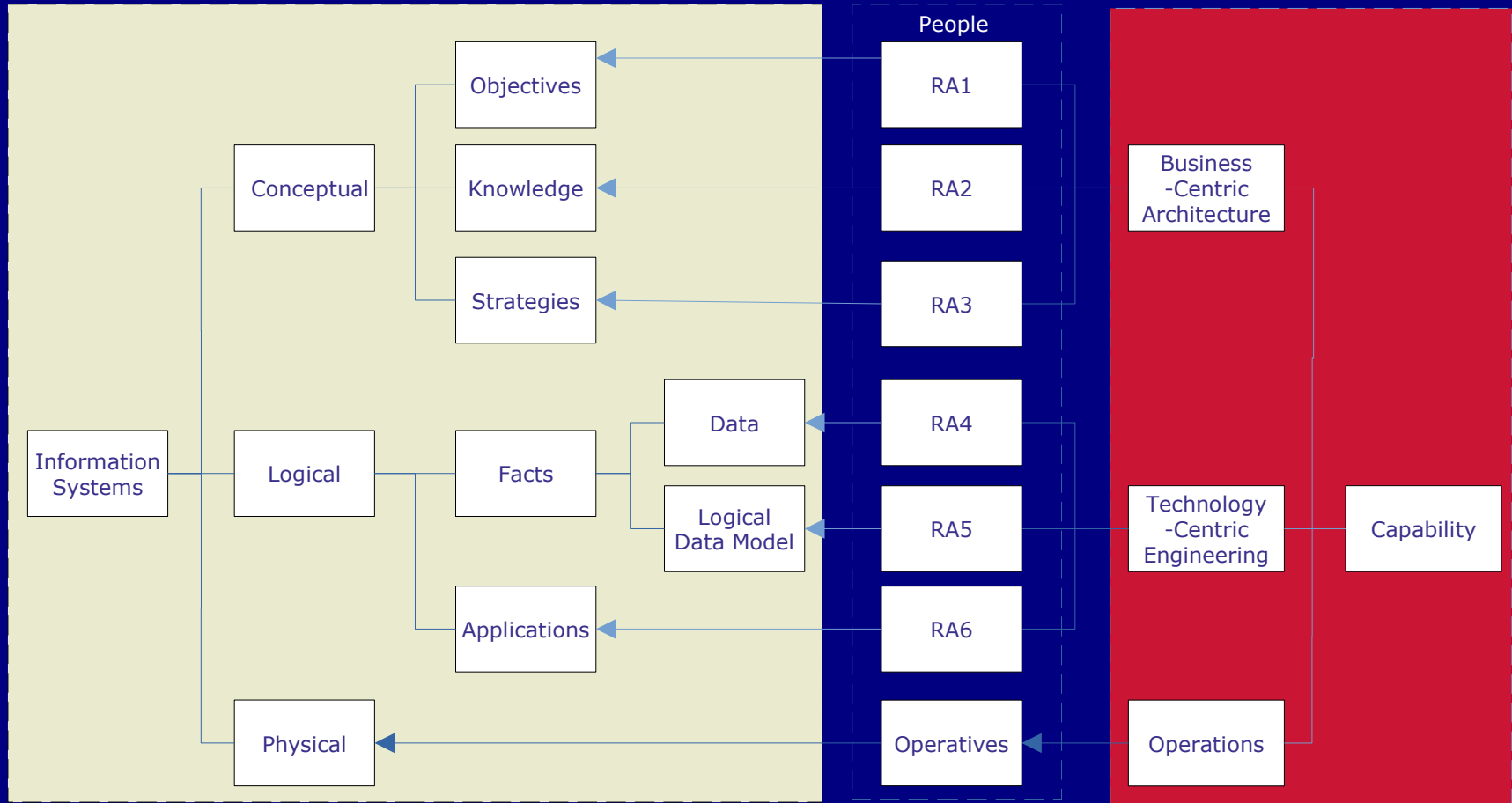


1.2. The Logical Ripose Architects: I will now resolve the 'many to many' relationship between 'Logical Systems' and 'People' and between 'Engineering Capability' and 'People' by introducing the Ripose Architect grades 4, 5 & 6.



A better solution

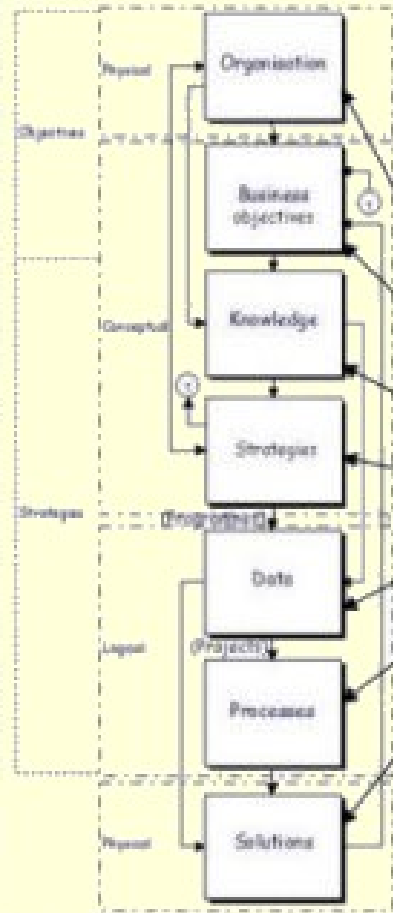
1.3. Operatives: I will now resolve the 'many to many' relationship between 'Physical Systems' and 'People' and between 'Operational Capabilities' and 'People' by introducing the 'Operatives'. These are all the rest of the stakeholders that the RAs need to work with in order to accomplish the task of 'Designing' a better enterprise. By this time all the 'Deliverables' have been resolved.



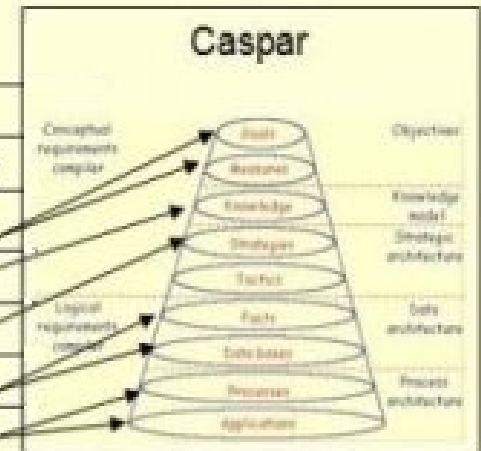
Welcome

To the Ripose training courses.
You are about to embark on a journey
like no other

Information architecture



Ripose architect	Equivalence
Grade 0	Information architect - Grades 1 through 6
Grade 1	Enterprise/business architect
Grade 2	Business knowledge architect
Grade 3	Strategic planner
Grade 4	Data architect
Grade 5	Solutions architect
Grade 6	Software engineer



Once you have completed this course you will be
able to tackle any approach and improve on their
processes and deliverables.