

UML enabling the Content Framework

Selvyn Wright

swright@celestial.co.uk

www.celestial.co.uk

+447778 449924

Agenda



- An introduction to modelling and little history
- Are we the first to struggle with modelling concepts
- Language and its importance
- What can we learn from Bob the builder
- Instance based modelling
- Meta modelling
- Meta-meta modelling
- From Bob the builder to the TOGAF Content Framework

Introduction

■ Modelling

- *a standard or example for imitation or comparison*
- *a representation, generally in miniature, to show the construction or appearance of something*
- *a style or design of a particular product*
- *a pattern or mode of structure or formation*
- to form or plan according to a model
- to give shape or form to; fashion
- to simulate (a process, concept, or the operation of a system), commonly with the aid of a computer

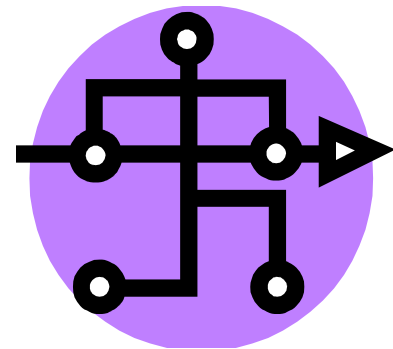


- *The above courtesy of dictionary.com*

Only fools go...

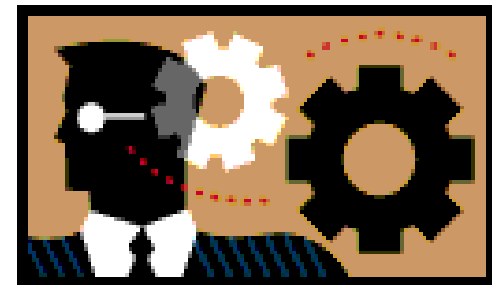


- “Only fools go to war without a plan...”
 - **The Art of War** by Sun Tzu
- “if the design of a product has been sufficiently studied, then changes in it will come very slowly”
 - **My life and work**, by Henry Ford
- “I draw a plan and work out every detail on the plan before starting to build”
 - **My life and work**, by Henry Ford



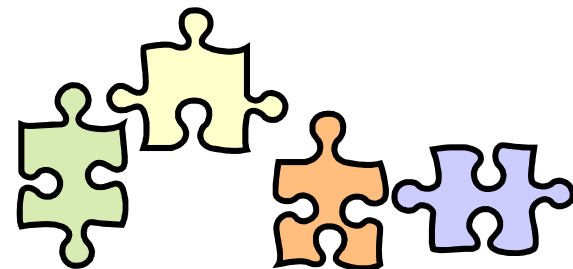
Are we the first...

- As a species we have been modelling for years
 - Henry Ford is a great example
 - The Egyptians needed a language to convey their ideas for the construction of the pyramids etc.
 - My mother used pattern to create dresses, the pattern was expressed in a form that she could understand
 - My brother who was a chef, he would use cookie type cutters to produce consumables
 - Biologist have been modeling for many years, as have chemist. Biologist used concept called classification
 - Building architects use a language to convey ideas
 - Einstein and Newton with their mathematical models



Importance of the language

- Models require a language and models are used to communicate ideas
- For a language to be effective it must have a grammar, the grammar defines how the components of the language can be assembled for different specifications
 - This is known as the meta-language (or meta model)
- Section 35.4.3 of TOGAF stresses the importance of a common language



What do I want



- When I look at any model I want them to be
 - Complete(ness)
 - Precise(ness)
 - Correct(ness)
 - Robust(ness)

- I want...
 - traceability

Bob the builder

- Bob has been in the industry for years



- Bob's problem is that for a quite a few of his jobs he has to use specialised tools



Bob's frustration

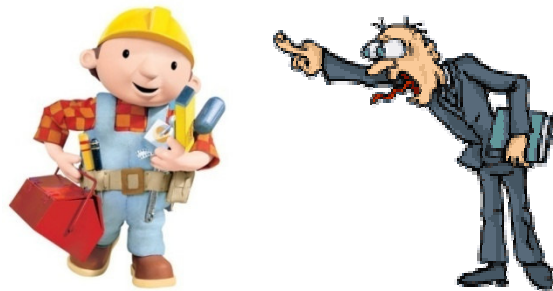


- Costly
- Can't always get the tool he wants
 - Too heavy
 - Too wide
 - Too narrow
- The good news is that if Bob uses one of these hammers, he can quickly employ someone else who might also be familiar with these tools

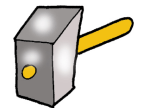
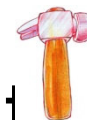


Bob's initial solution

- Make the tools himself
 - There is a time cost and error prone

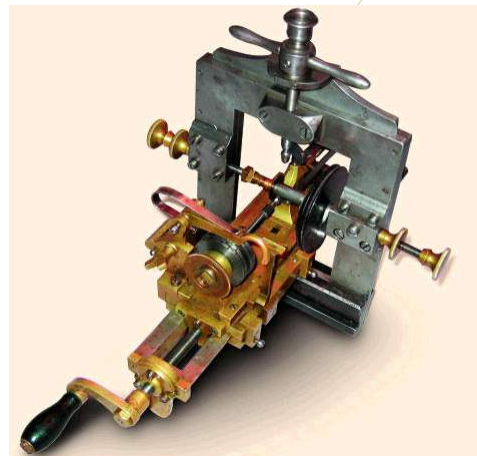


- Bob starts to see frustrated customers
- Bob's new tools are not consistently built
- Each tool has its own quirk



Bob's new solution

- Build a tool to build the tools

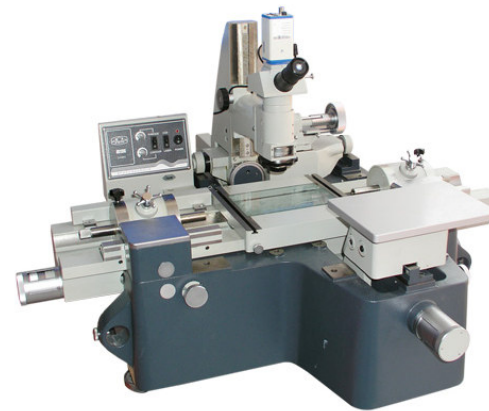


- Consistently build tools ie. a consistent approach
- Reduce time to build tools



Bob sees himself as a visionary

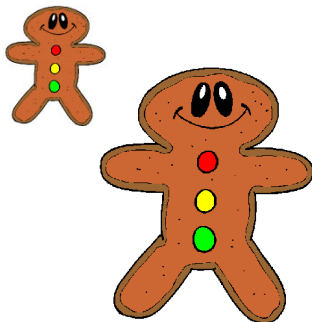
- Bob realises that if he can be build a tool make tools
 - Maybe he can make a tool to make the tool to make the tools



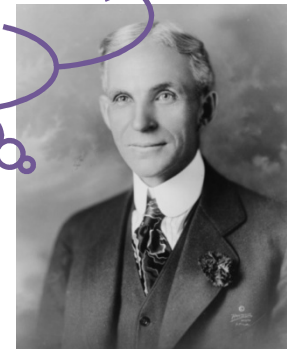
- So Bob uses his first set of tool making kit to make his “tool for tool making”

Lessons from Bob

- Well, these aren't really lessons from Bob but from Henry Ford
- Henry Ford spent many years creating the individual parts he needed for his visionary car
- He later spent many years creating the tools that he would use to create the parts he needed for his visionary car i.e. a cookie cutter

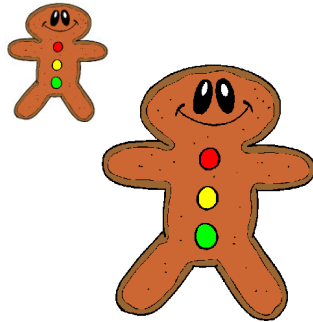


So maybe, I could
create a tool to make
the parts

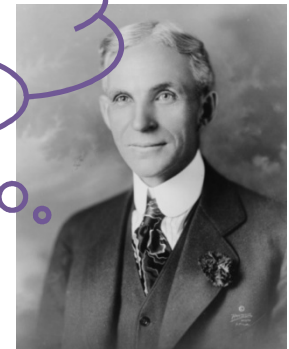


Moving to meta-models

- But our example is too fixated on the specific type of “thing”
 - Like Bob, Henry Ford found himself spending a lot of time creating tools to create parts
 - A concept known as tool-setting
- But what if we could make the tool to create the tools
 - A mechanism to create tools for different situations



So maybe, I could
create a tool to create
the tools

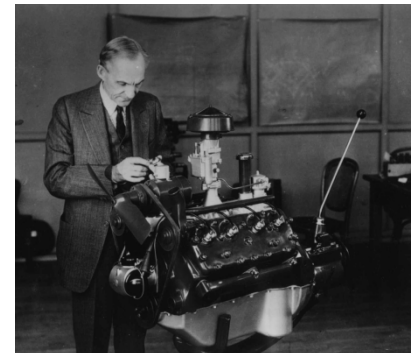


Successful meta-models



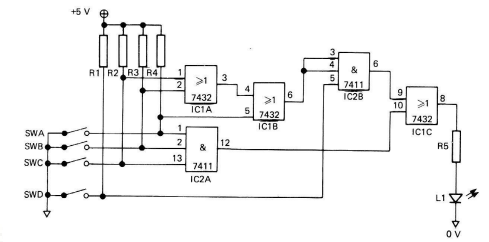
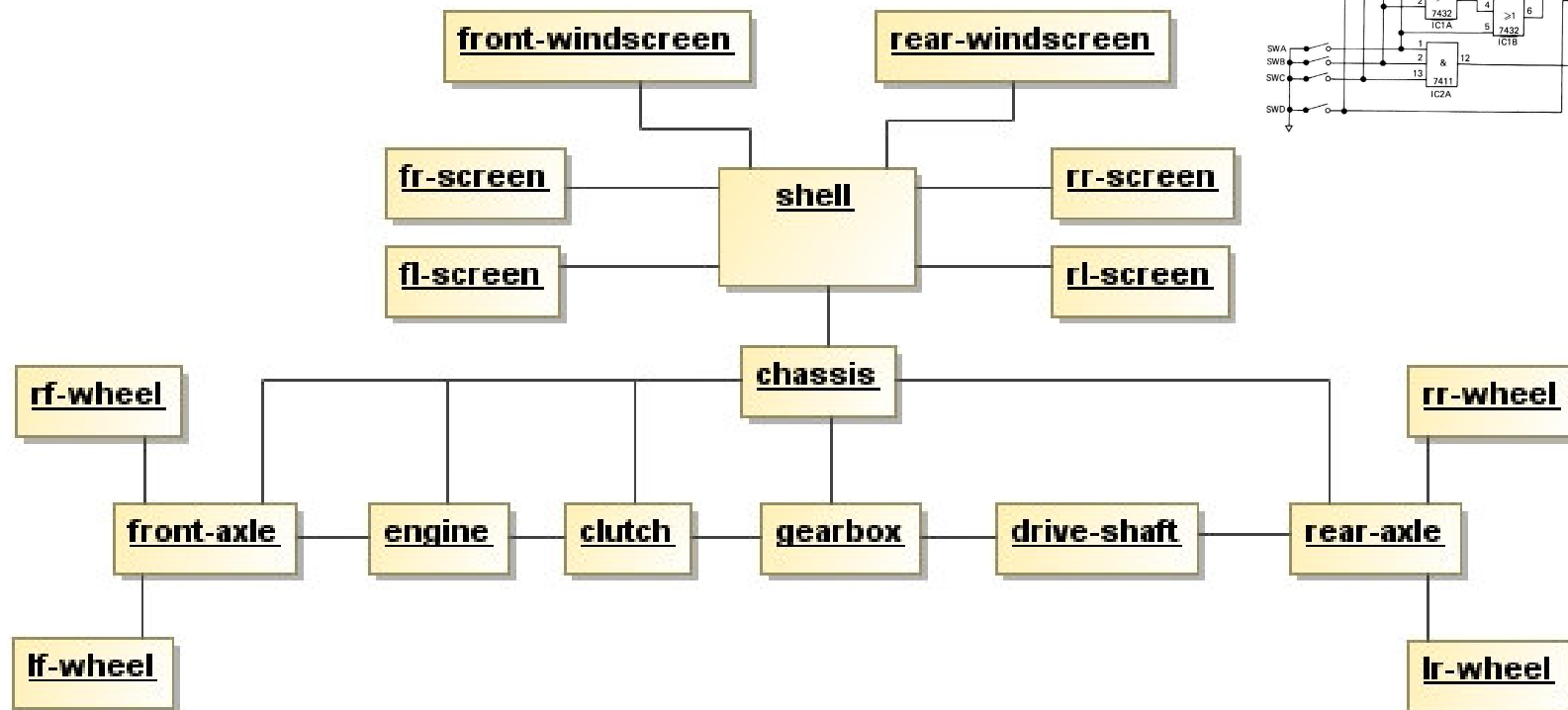
■ IP realization

- The more economical methods of production did not begin all at once. They began gradually, just as we began gradually to make our own parts...
- The great economies began in assembling and then extended to other sections so that, while today we have skilled mechanics in plenty, they do not produce automobiles, they make it easy for others to produce them. Our skilled men are tool makers, machinists and pattern makers.



An instance/unclassified model

- An example: the parts of a car

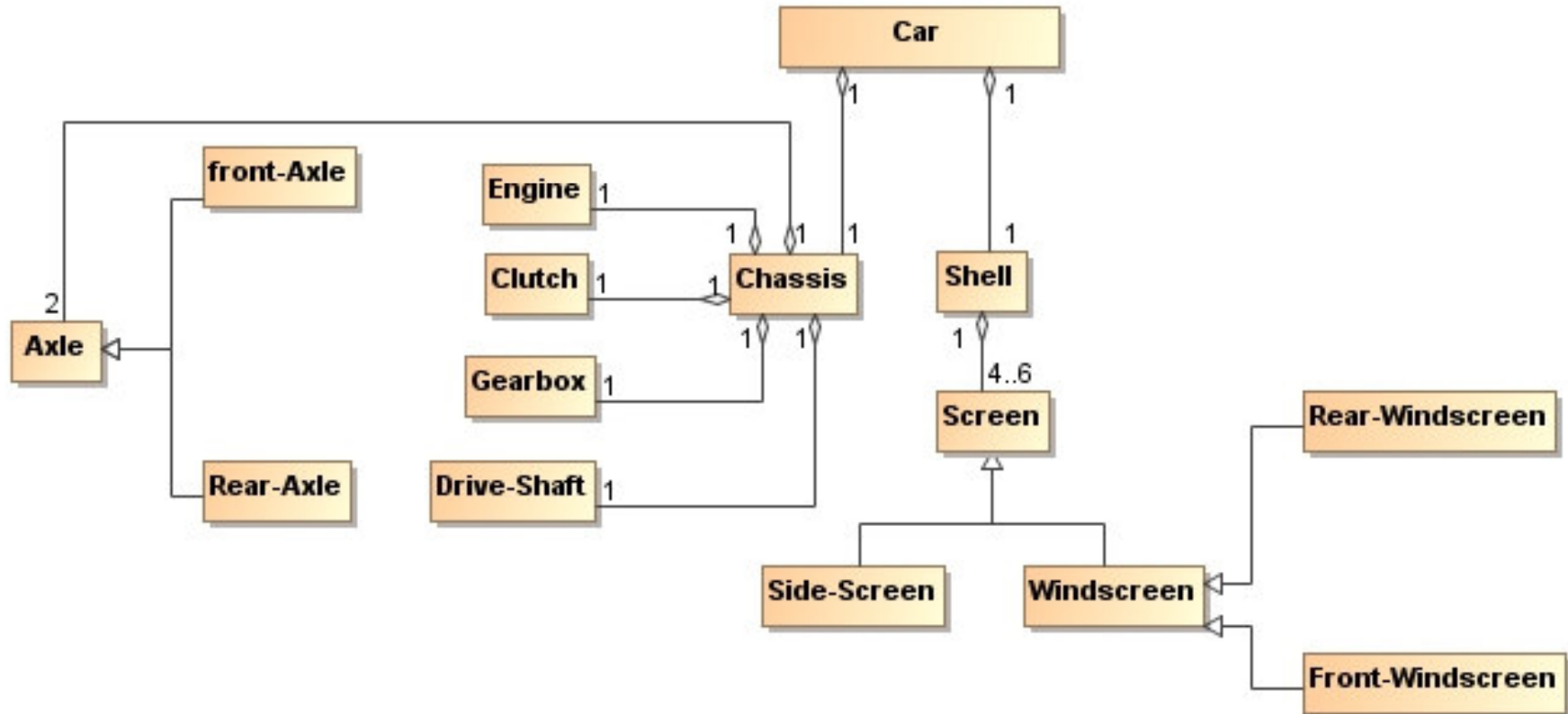


- “can we classify these parts?”



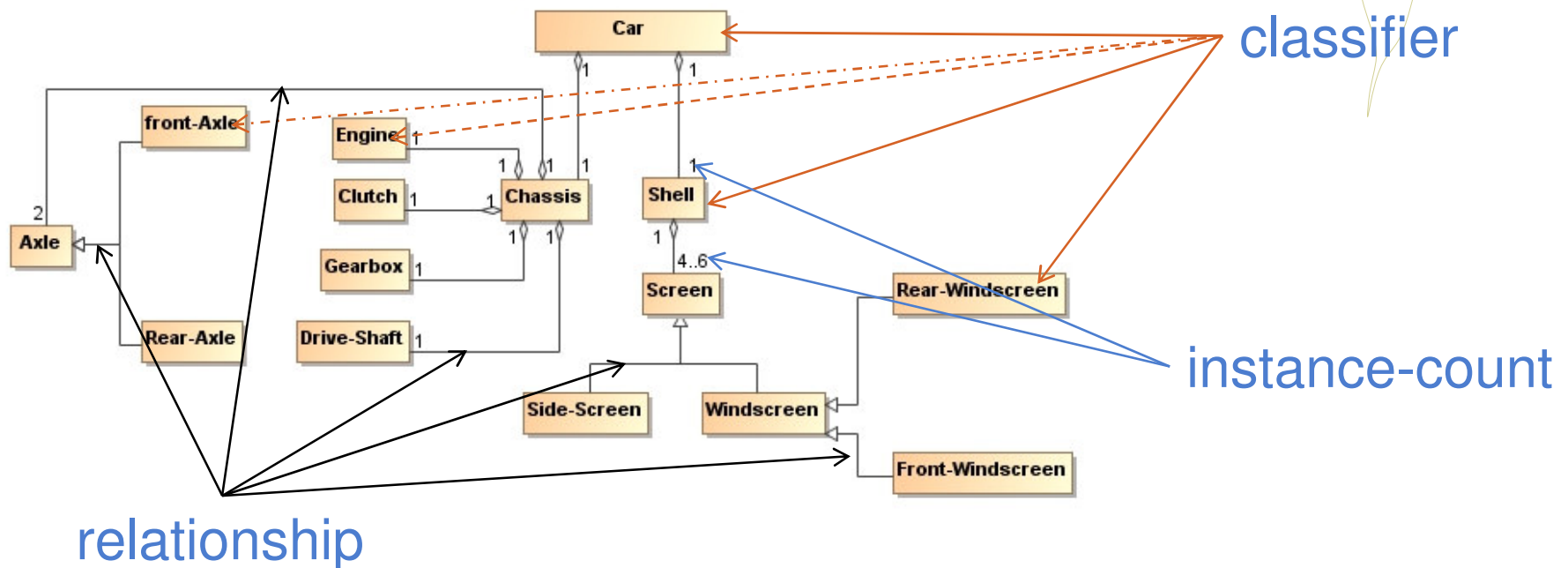
Moving to a classification model

- The previous model can be redrawn using classification, nothing new here!



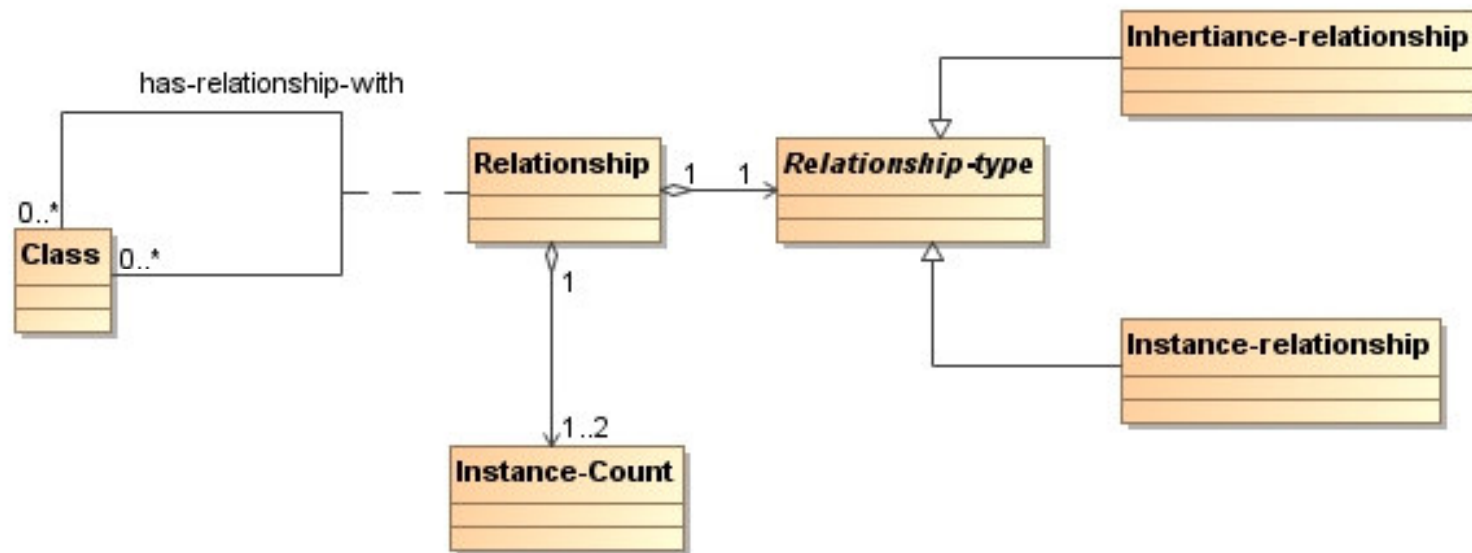
Moving to meta-models

- A meta-model describes how a model may be constructed
 - It specifies a notation that can be used in many situations i.e. meta-language



Moving to meta-models

- It specifies a notation that can be used in many situations i.e. meta-language

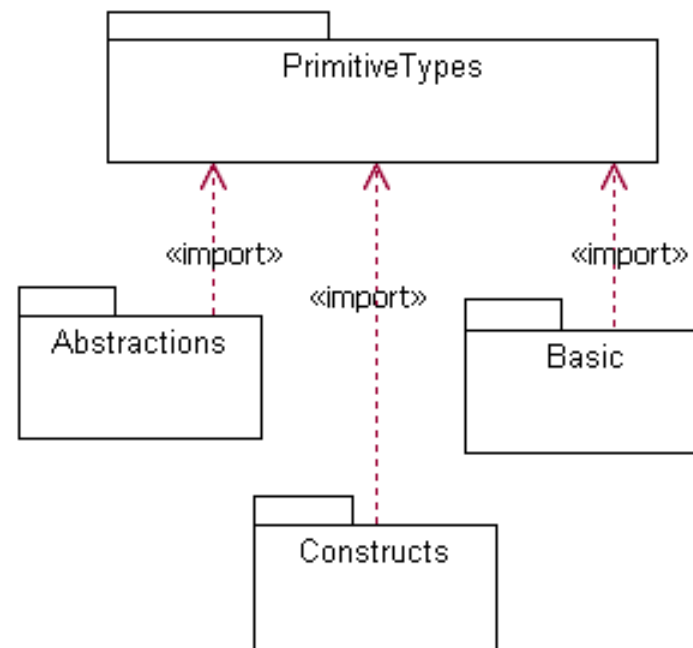


- The typical role of a meta-model is to define the semantics for how model elements in a model get instantiated.

UML understanding of this problem

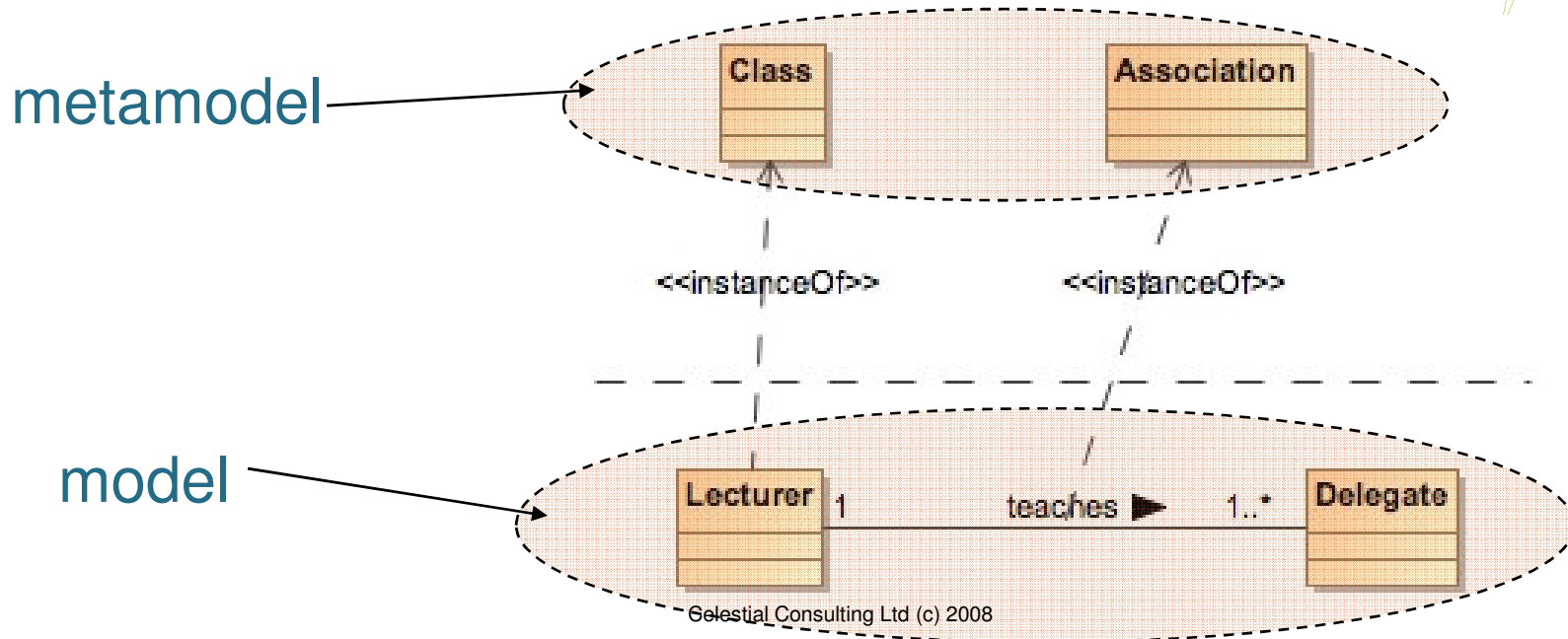


- The authors of UML understood that all the issues that Bob and Henry had to tackle were common to the software/IT industry
- At the core of the language architecture is a “core model”
 - See UML infrastructure



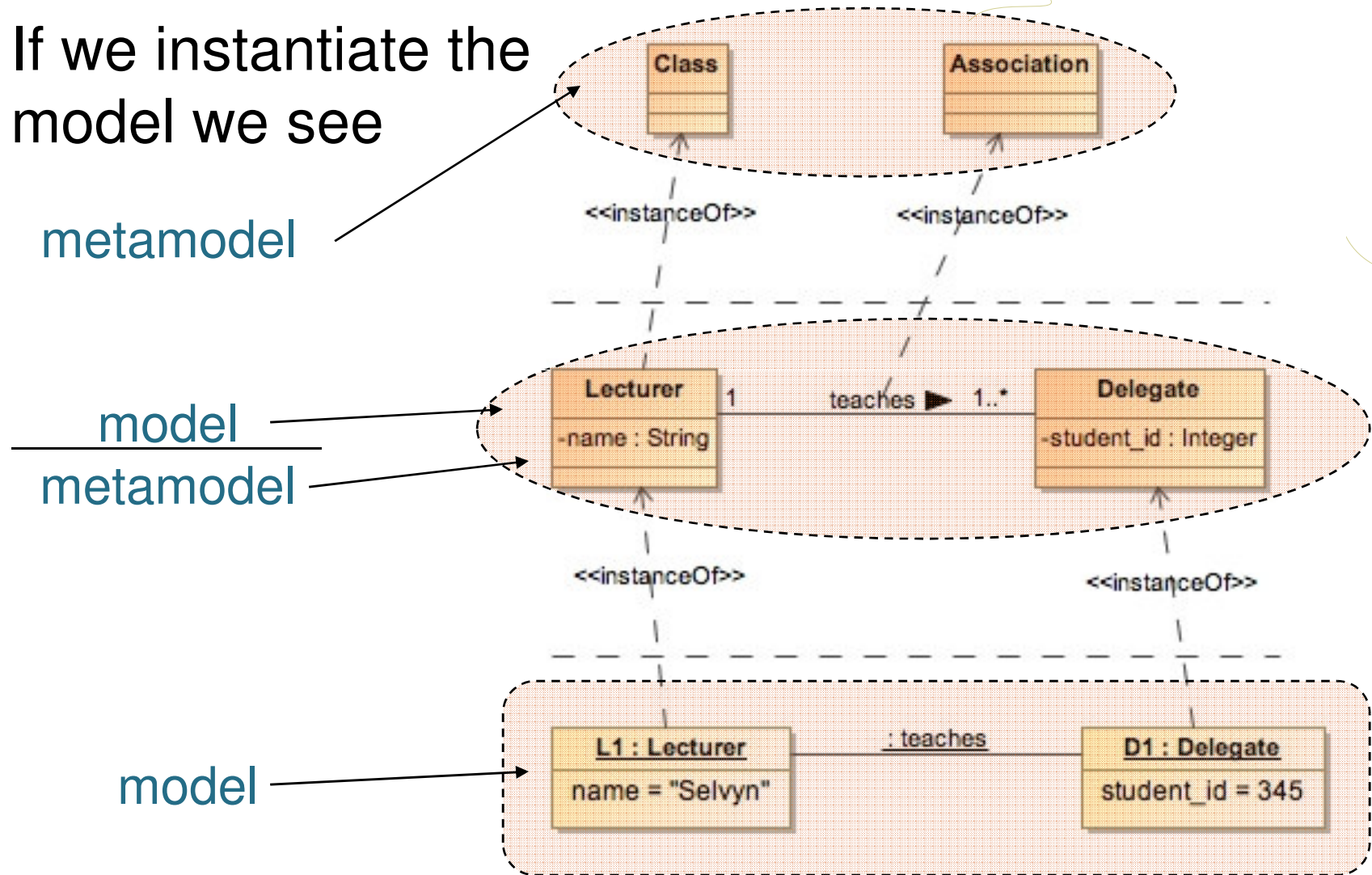
UML meta-model

- All UML meta-model is instantiated from the meta-meta-class model as defined in the InfrastructureLibrary
- The UML meta-model imports and specialises meta-meta-classes in the InfrastructureLibrary



UML meta-model part II

- If we instantiate the model we see



UML meta-metaclassification



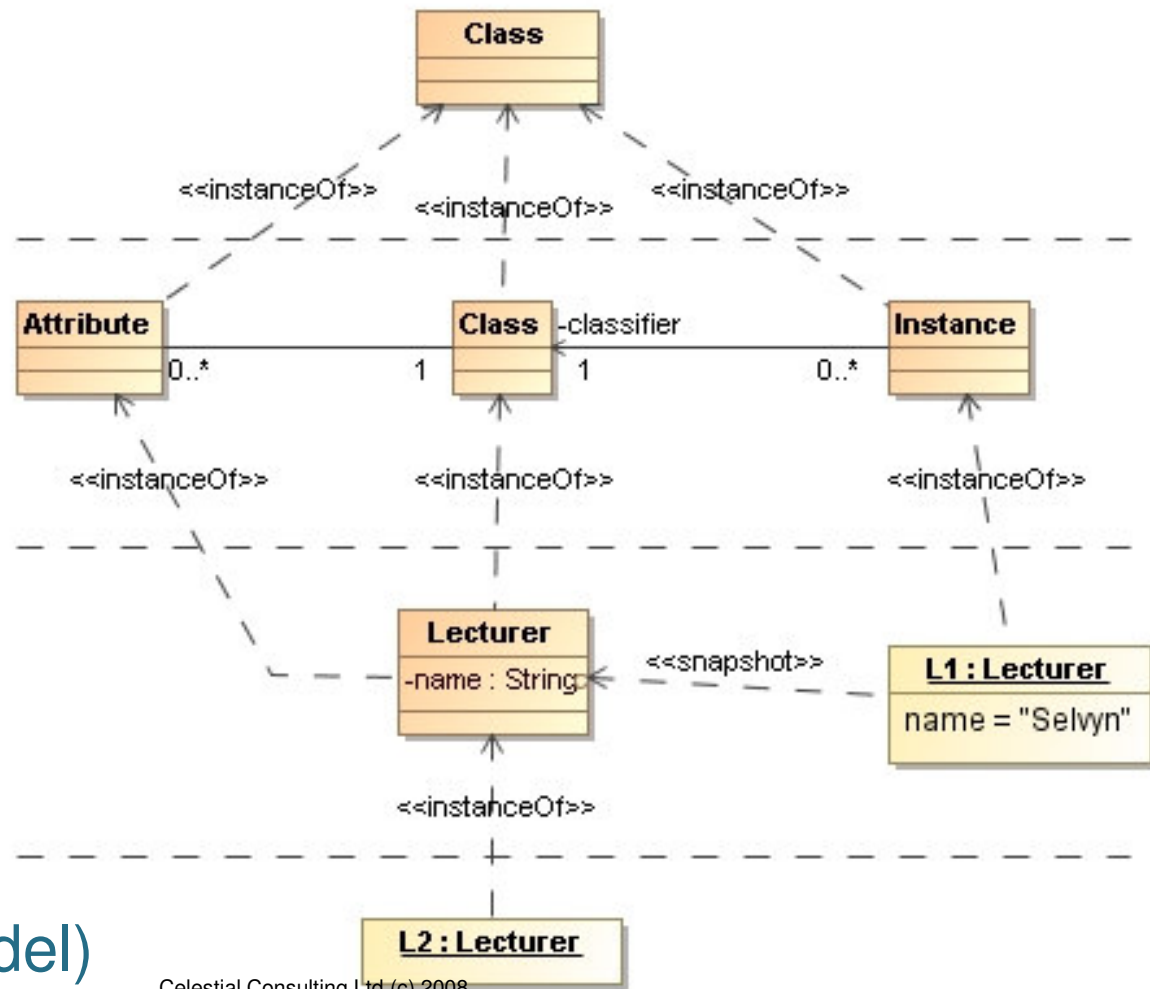
- Let's look at the problem from another angle

M3 (MOF)

M2 (UML)

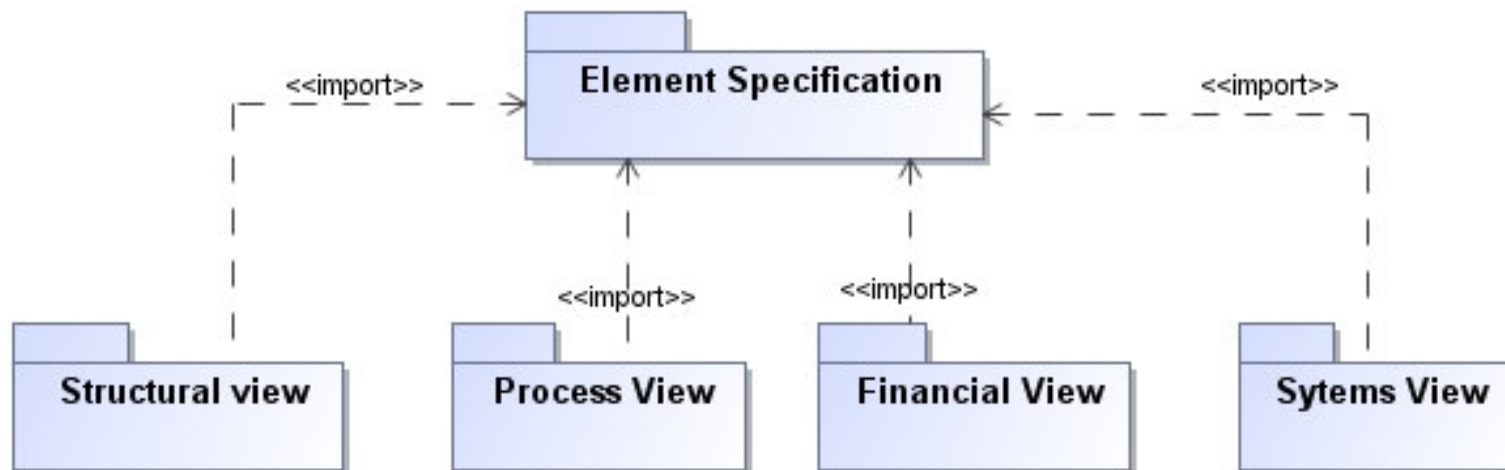
M1 (User Model)

M0 (Runtime Model)



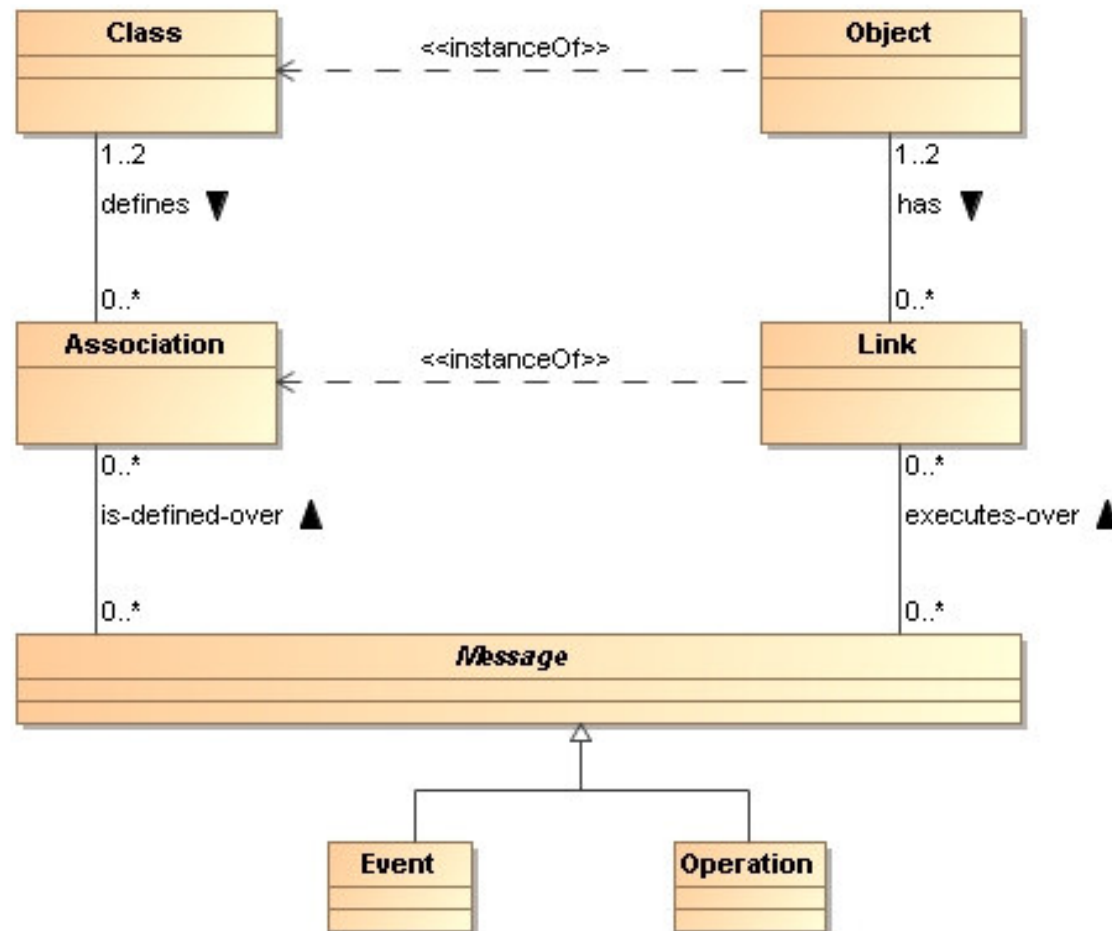
meta-meta models benefits

- Defining the elements in a view independent manner allows them to be reused across multiple views {meta meta specification}
 - Each view could then utilise these elements in a manner that is constrained by the view {the meta-language}
- The meta-meta language brings to the table a consistent language; *clear syntax and semantics*



Reuse across models

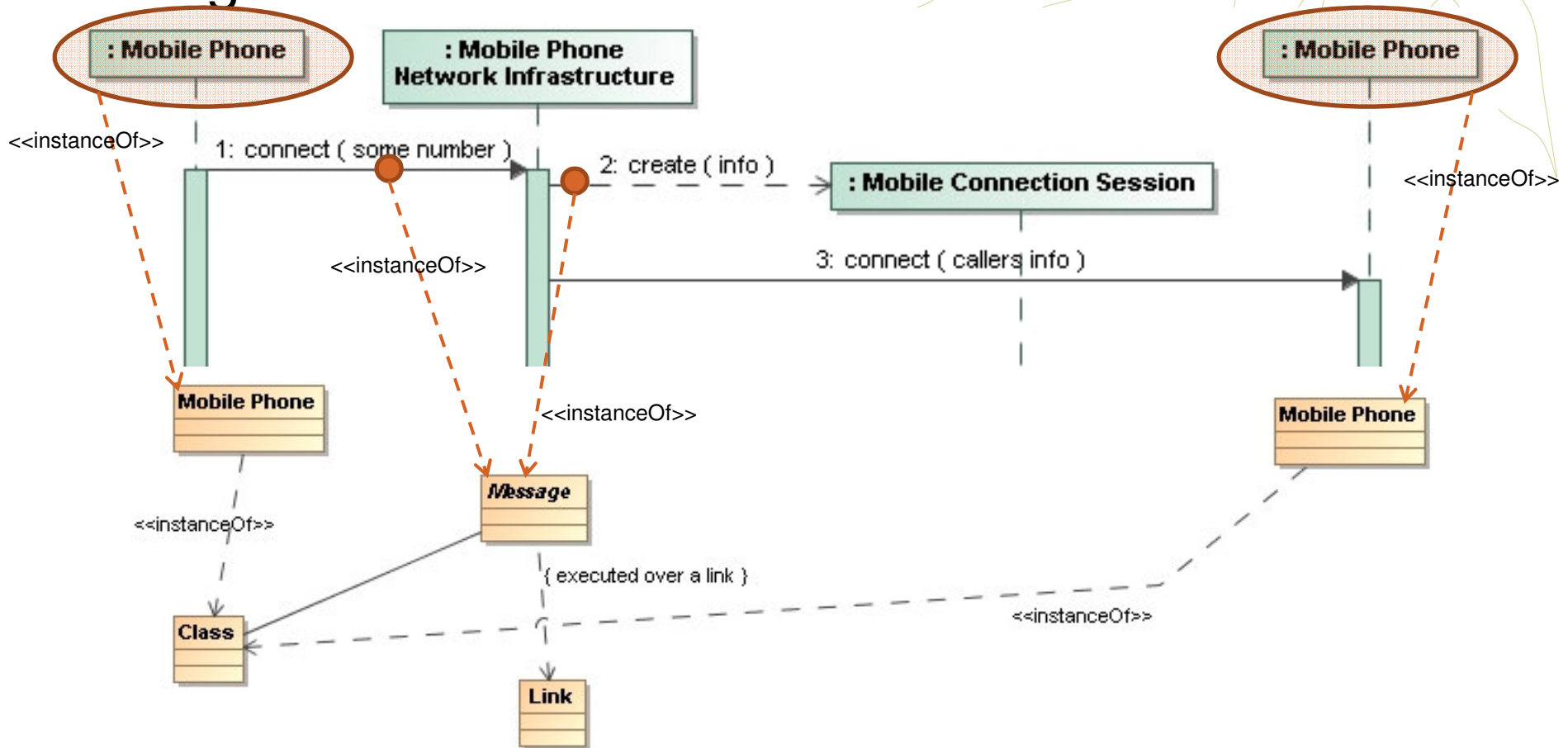
- This is a snapshot of the relationship between classifiers, objects, messages, associations and links



Communications view to Structural view



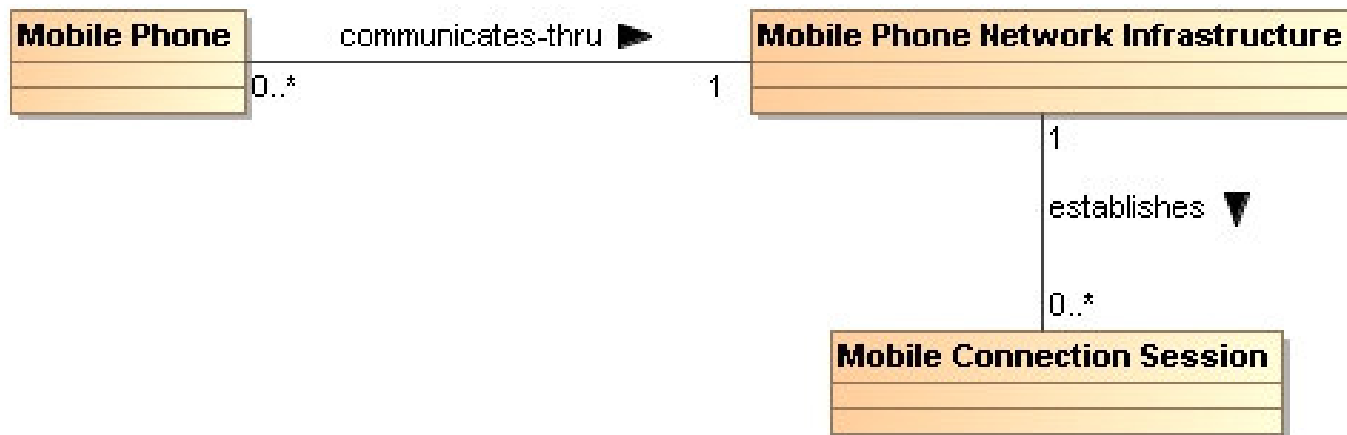
- Consider the following simplified communications diagram



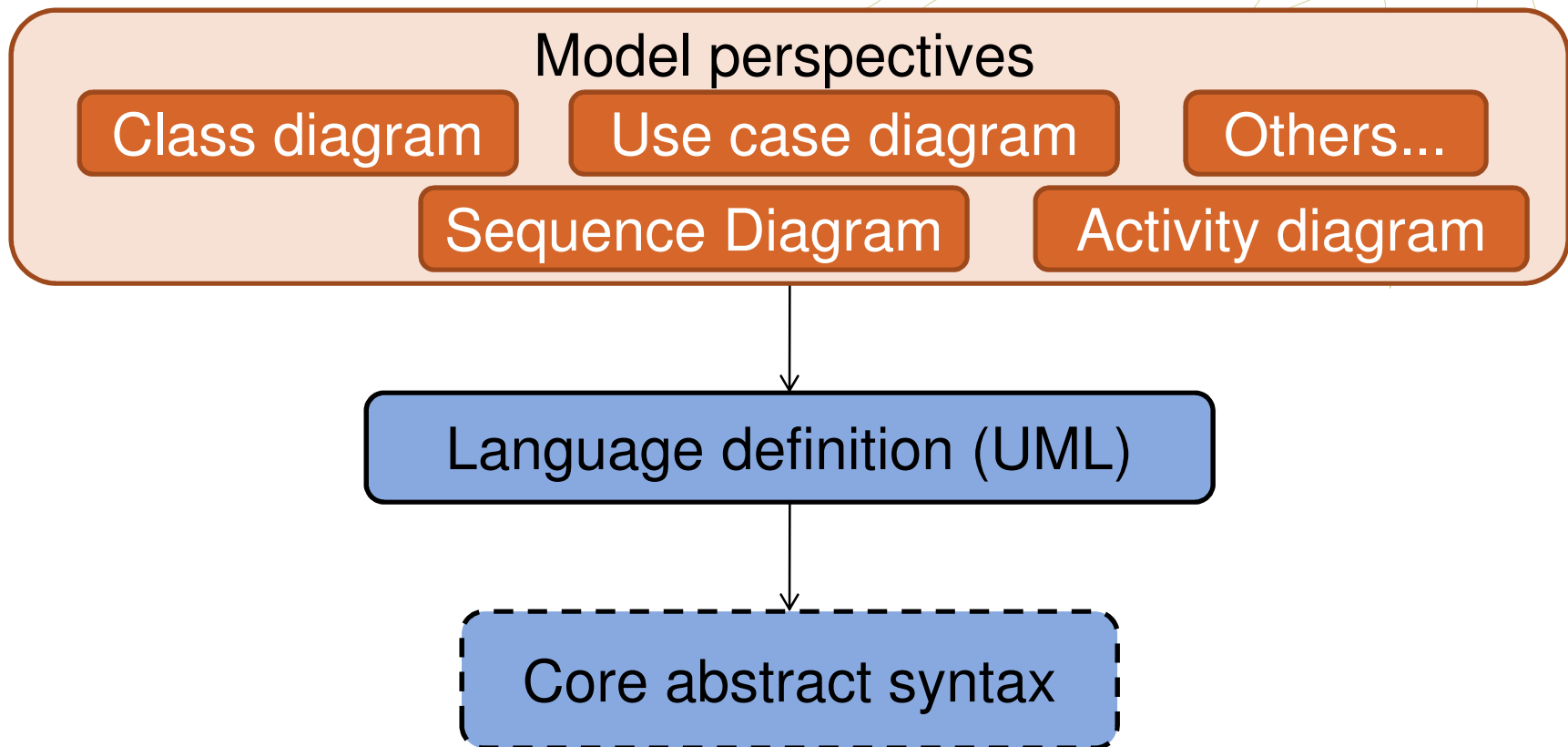
Communications view to Structural view



- Leading on the previous example and using the meta-model rules, it is now possible to construct a structural view of our world



So what does this mean (UML speak)



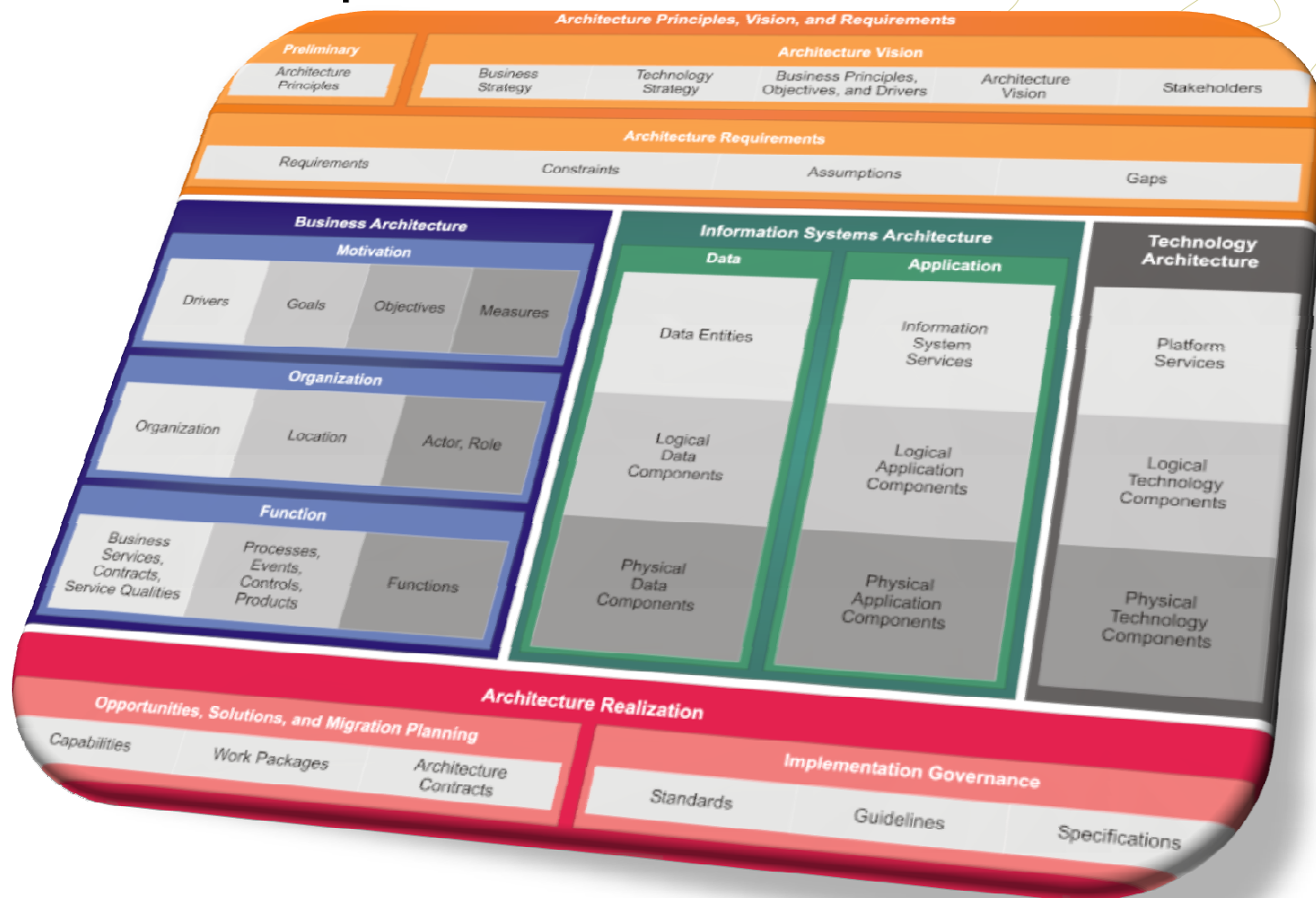
TOGAF Content Framework



- An attempt to create a consistent language across all aspects of EA work
- An ontology
 - Defining the elements and their relationships
- Like any language it needs to be extensible but robust to allow it to be used in numerous situations
- Elements need to be well defined (clear semantics)
- Relationship between elements need to be well defined (clear syntax and semantics)

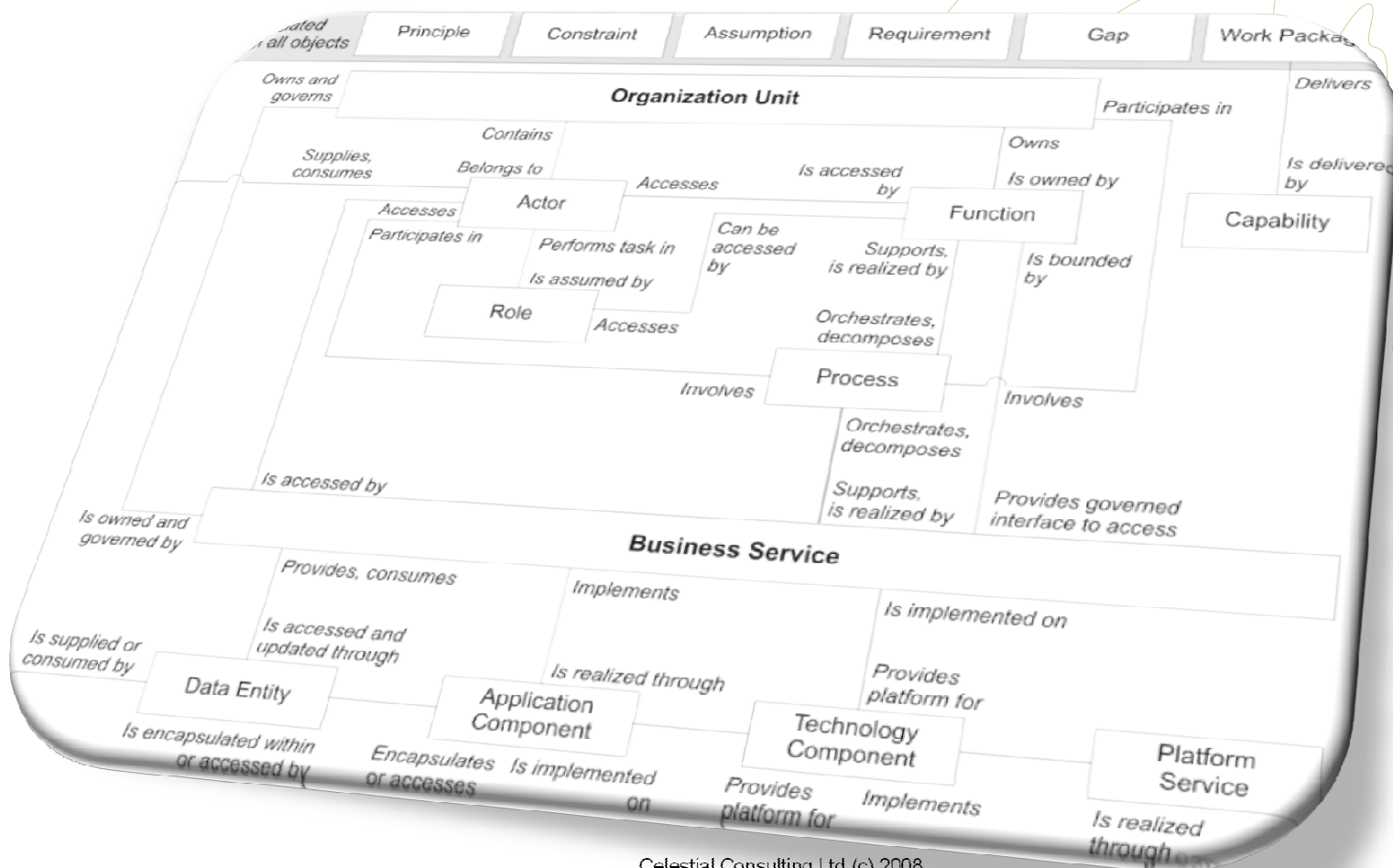
Recap of some of the key views

- Detailed representation of the content metamodel



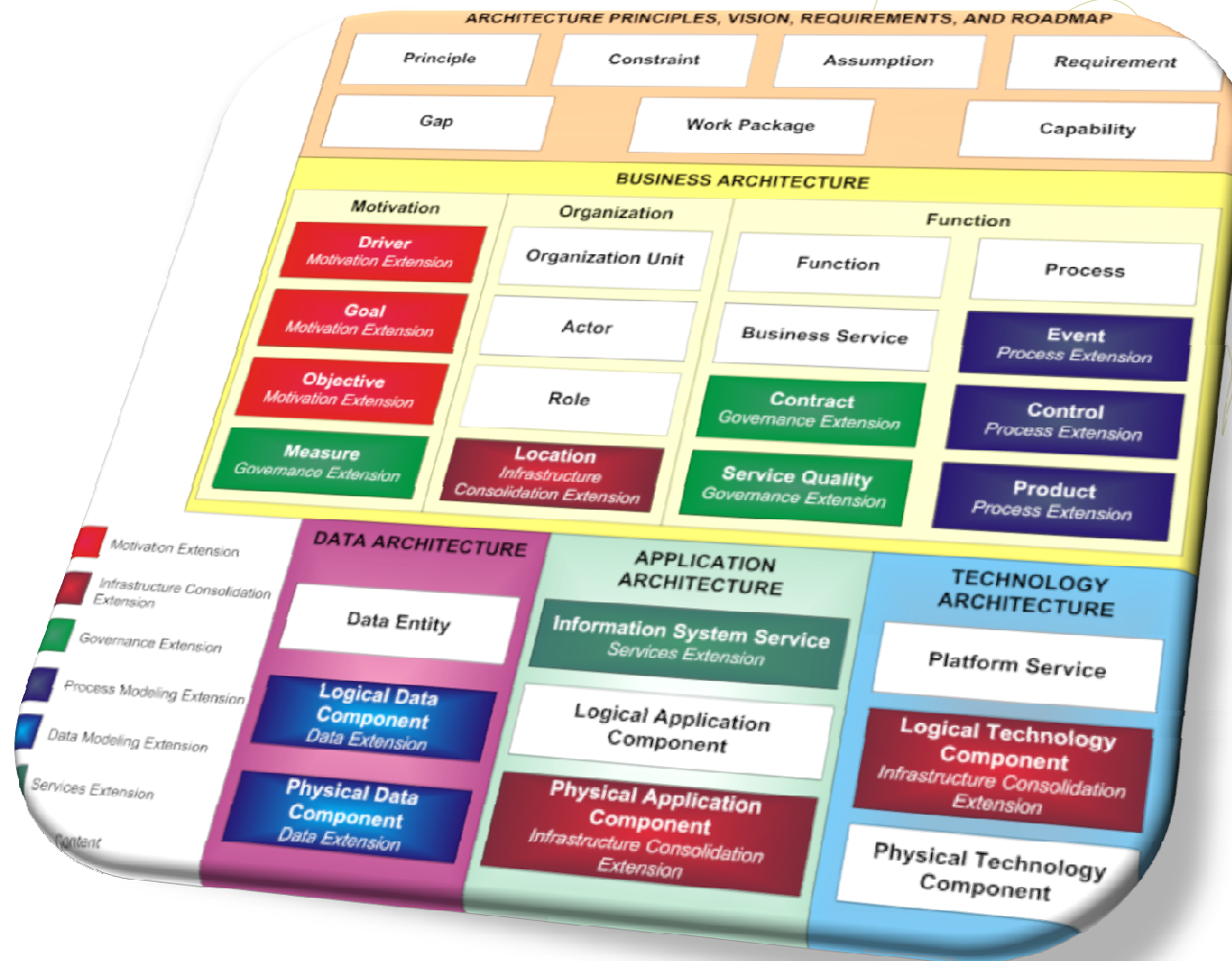
Recap of some of the key views

- Core entities and their relationships present within the Core metamodel



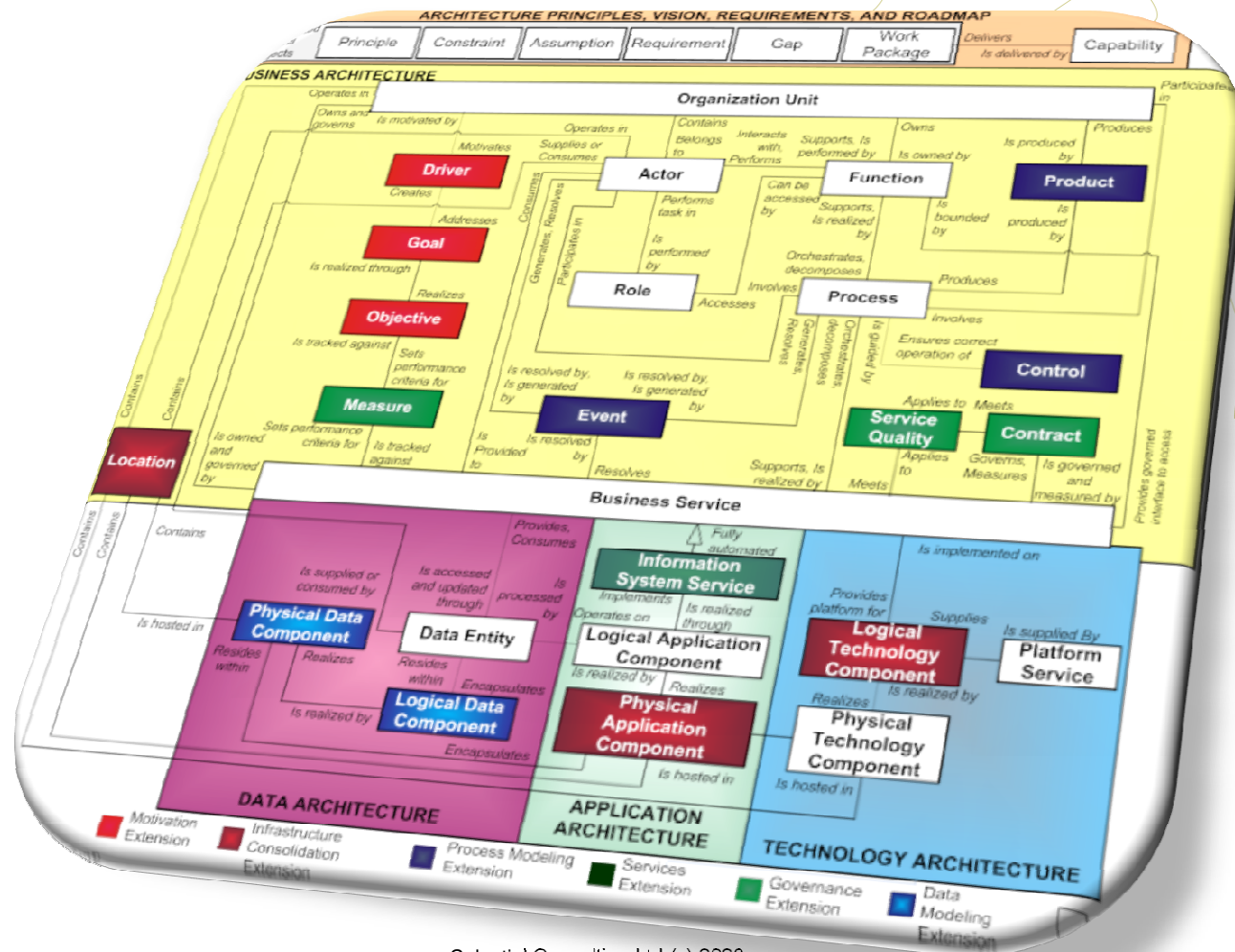
Recap of some of the key views

- Content metamodel with extensions



Recap of some of the key views

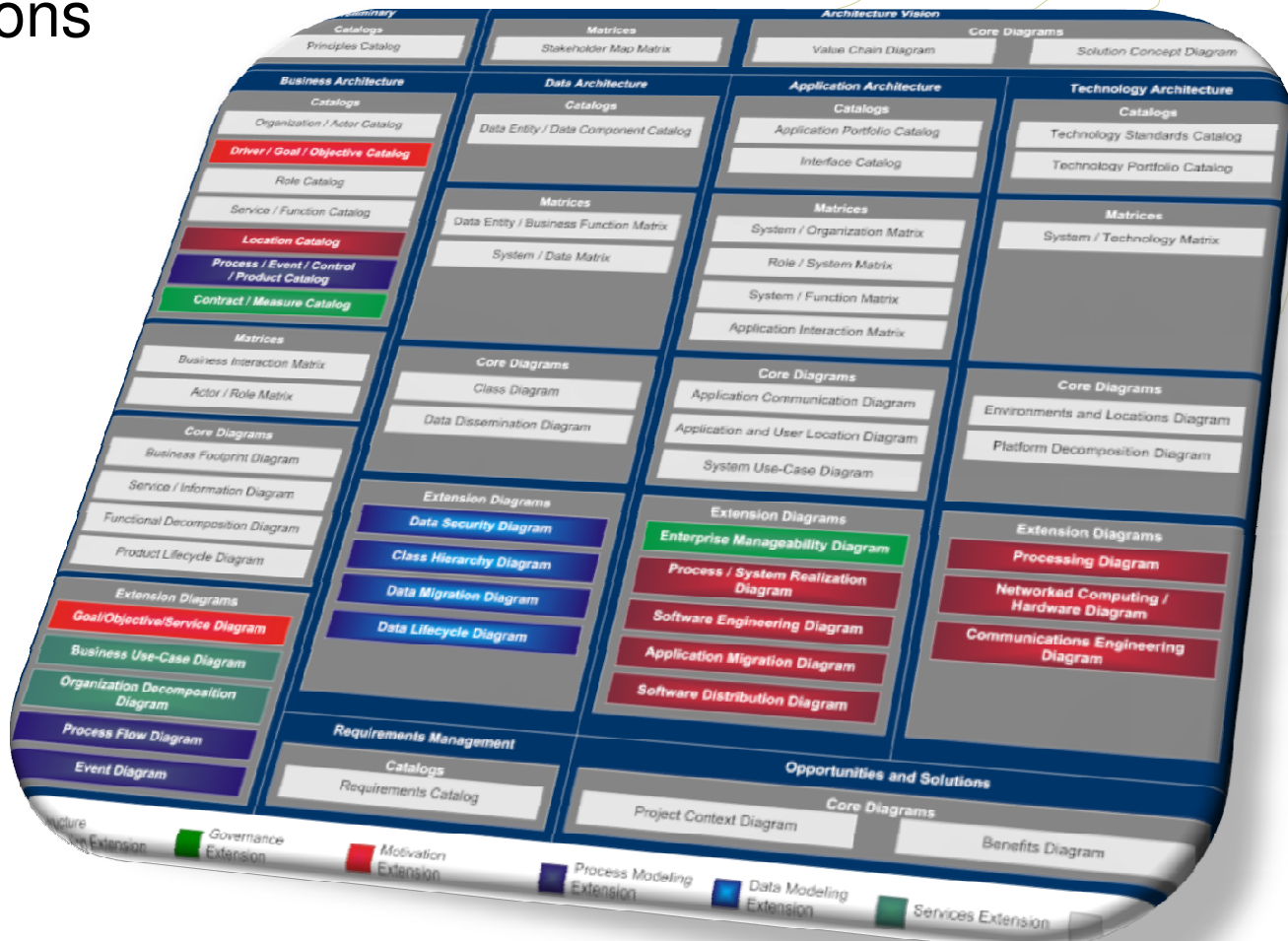
- Relationship between entities in full metamodel



Recap of some of the key views

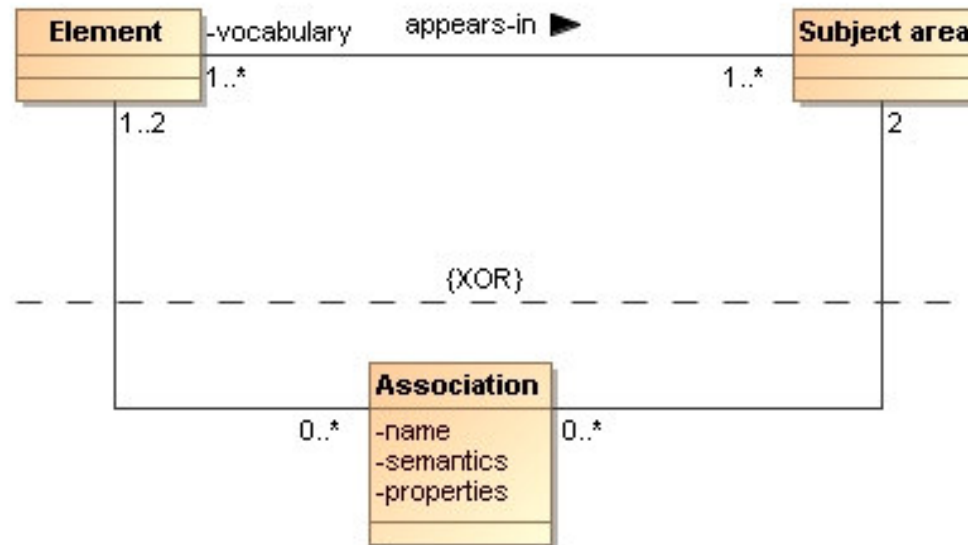


- Viewpoints associated with Core Content Metamodel and Extensions



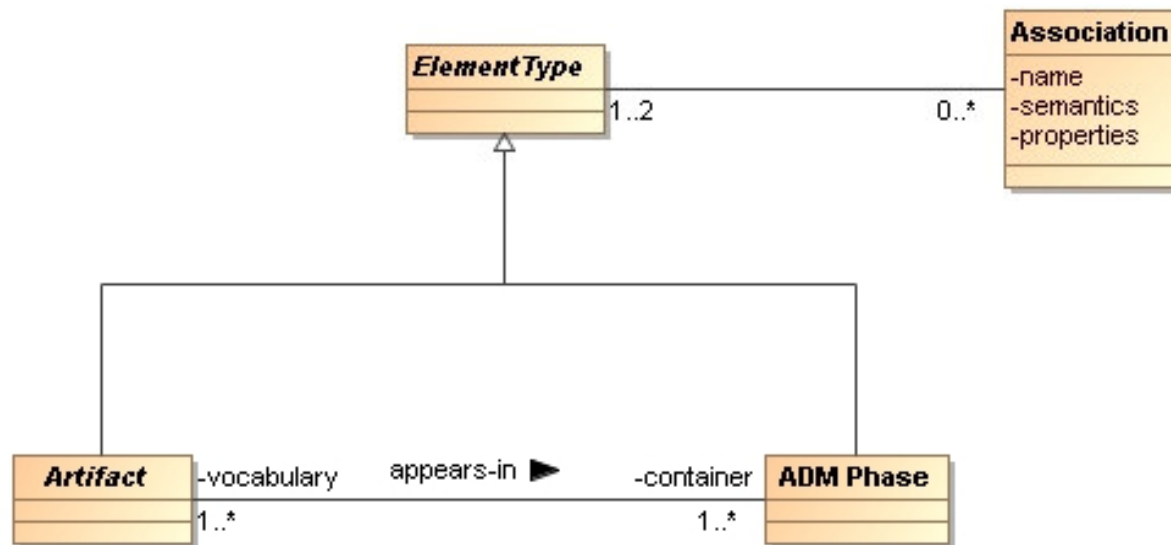
Content framework elements

- Can be defined as



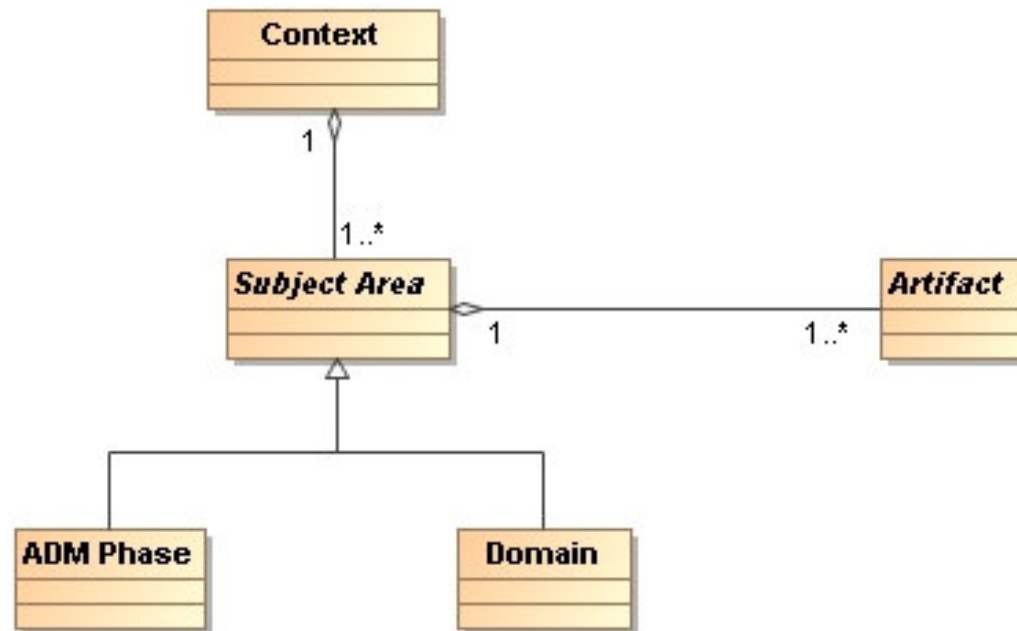
Content framework elements

- But a refinement of the previous model is



Content framework elements

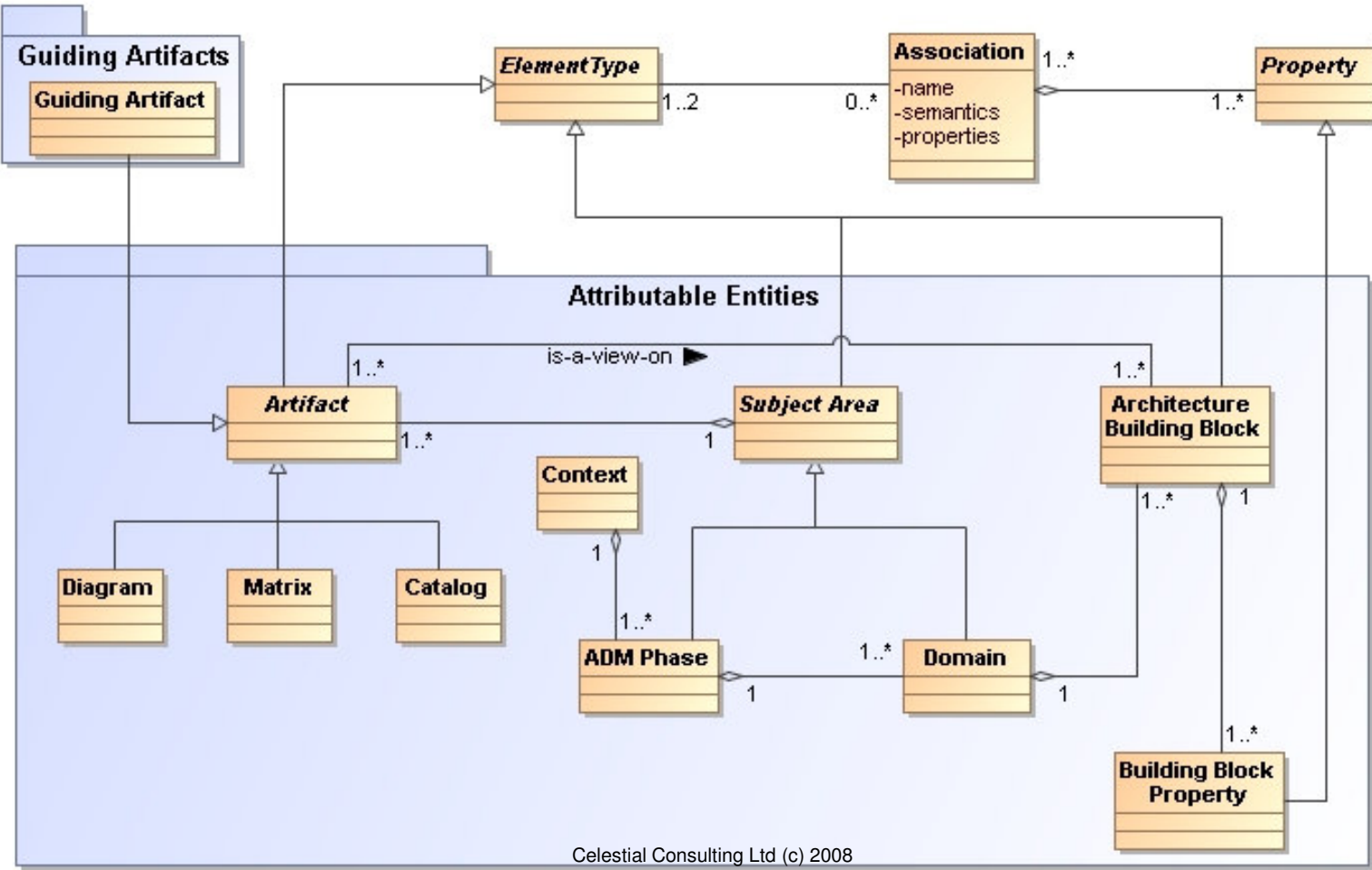
- On closer examination of the content framework and early discussions with Content Framework authors we saw



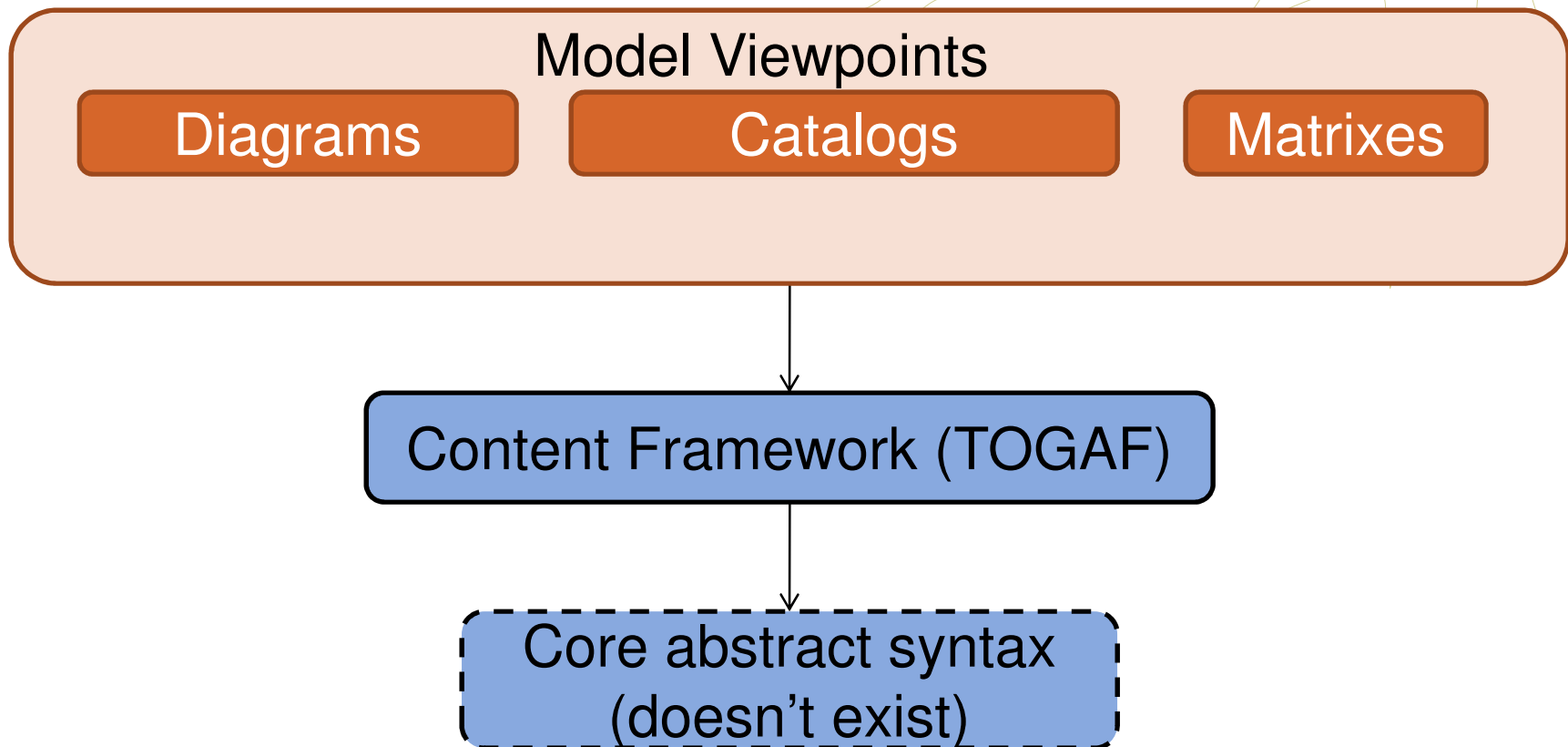
Draft meta-metamodel of Content FX



- After more discussions with content fx author and examining the content fx, we could concluded with something like this



So what does this mean (TOGAF speak)



Summary



- We have seen that modern 21st century man isn't the first to attempt modelling
- We can learn a lot from people like Bob the builder and Henry Ford
- The importance of language and modelling
- Moving from instance to class to meta-class to meta meta-class based modelling is where the IP is held and realised
- Using meta metaclassification should be consistency across all models which leads to traceability

Questions



- Thank you!
- Selvyn Wright
- swright@celestial.co.uk
- +447778 449924

