

# SKA CONSORTIA MODEL – AN OVERVIEW

Daniel Hayden

# Contents

- Definition
- History
- Current state overview
  - Participants
  - Administration
  - Model content
- 3 options for a collaborative model
- Model structure – diagram, modules explained, implications
- SE2 framework
- Advantages
- Costs

# Definition

**SKA consortia model** – a single SysML model of the SKA system, from telescope level down to sub-Element level, being developed by SKAO and consortia modellers

# History

- **Time:** Process began in March/April 2015
- **Rationale:**
  - Several consortia were already doing SysML modelling
  - Creating a separate system model would mean having to manually update content from consortia models to system model at intervals
  - To avoid this, a single model framework was proposed
- **Early participants:** CSP, TM, SKAO

# Current state overview - participants

- **Facts:**
  - ~12 modellers from 5 consortia (LFAA, TM, SDP, SADT, CSP) and the SKAO
  - For DSH, INFRA SA, INFRA AU, AIV – some surrogate modelling done by SKAO
  - Participants comprise a working group that has videocons every 2 weeks
- **Therefore** significant participation, **but** still lacking substantial telescope level and direct consortia level activity

# Current state overview – administration



- **Model hosted** on Cameo teamwork server
  - Server owned currently by SKA SA
  - License should soon allow for unlimited number of concurrent connections
  - Intention to shift ownership of model in future to SKAO

# Current state overview – model content



Element category	Types in category	Approximate number
Structure	Block, SKA product, SKA product family, SKA flow, actor	900
Interface	Interface block, SKA interface	500
Behaviour	Activity, interaction, state machine, state, use case	1400
Requirement	Requirement, SKA L1 requirement	2200 (but this is due to a mass import)

# Current state – model content

- Therefore:
  - significant content, **but** there is almost no configuration information (reference, author, date). This is an obstacle to using the model content ‘formally’
  - structures, functions, interfaces, requirements are being modelled, **but** a consistent SE methodology for linking these across the model is still premature



# An aside: 3 options for a collaborative model (citing G. Le Roux)



Option	Some pros	Some cons
1. One integrated, hierarchical model, from telescope to sub-Element level. Split into modules.	<ul style="list-style-type: none"><li>- Can do full traceability studies</li><li>- Consistency of definitions</li><li>- Separation of concerns</li><li>- Not too large to handle</li></ul>	<ul style="list-style-type: none"><li>- Cross-cutting work is more difficult / tedious</li></ul>
2. One integrated, hierarchical model, from telescope to sub-Element level. Monolithic.	<ul style="list-style-type: none"><li>- Can do full traceability studies</li><li>- Consistency of definitions</li><li>- Cross-cutting work is easier</li></ul>	<ul style="list-style-type: none"><li>- Single module becomes too large to handle</li><li>- No separation of concerns</li></ul>
3. One integrated telescope-level only model. Consortia models are separate.	<ul style="list-style-type: none"><li>- Easier implementation</li><li>- Separation of concerns</li></ul>	<ul style="list-style-type: none"><li>- Cannot do full traceability studies</li><li>- Less consistency of definitions</li></ul>

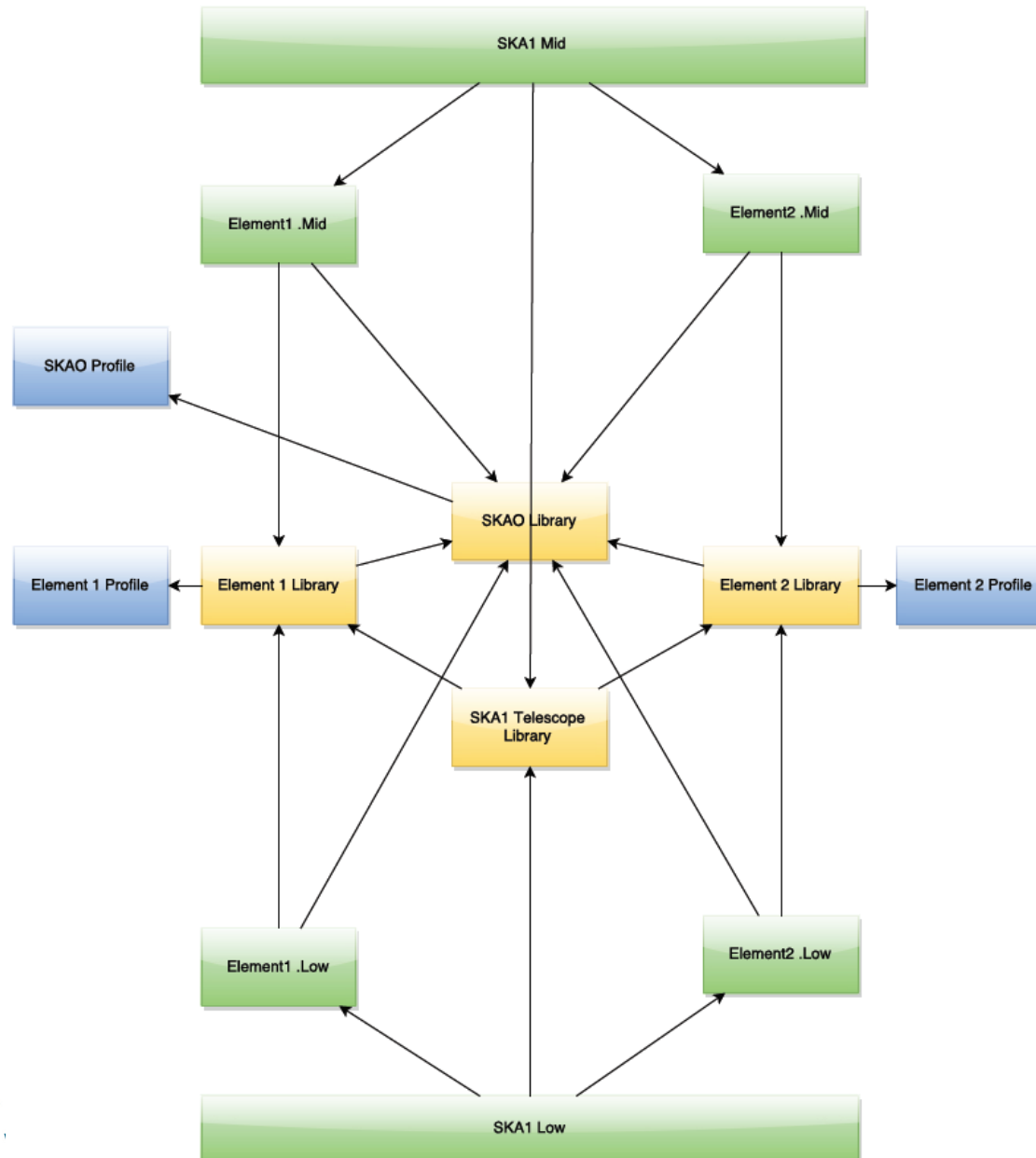
# An aside: 3 options for a collaborative model (citing G. Le Roux)



Option	Some pros	Some cons
<p>1. One integrated, hierarchical model, from telescope to sub-Element level. Split into modules.</p> <p><b>CONSORTIA MODEL</b></p>	<ul style="list-style-type: none"><li>- Can do full traceability studies</li><li>- Consistency of definitions</li><li>- Separation of concerns</li><li>- Not too large to handle</li></ul>	<ul style="list-style-type: none"><li>- Cross-cutting work is more difficult / tedious</li></ul>
<p>2. One integrated, hierarchical model, from telescope to sub-Element level. Monolithic.</p>	<ul style="list-style-type: none"><li>- Can do full traceability studies</li><li>- Consistency of definitions</li><li>- Cross-cutting work is easier</li></ul>	<ul style="list-style-type: none"><li>- Single module becomes too large to handle</li><li>- No separation of concerns</li></ul>
<p>3. One integrated telescope-level only model. Consortia models are separate.</p>	<ul style="list-style-type: none"><li>- Easier implementation</li><li>- Separation of concerns</li></ul>	<ul style="list-style-type: none"><li>- Cannot do full traceability studies</li><li>- Less consistency of definitions</li></ul>

# Model structure

This is  
option 1  
from  
previous  
slide



Each arrow  
represents a  
dependency:  
what a  
module can  
'see'

# Some modules explained

SKA1 Tel

- Telescope level module, for containing telescope level structures, behaviours, requirements. Can 'see' all its Element level modules and everything they can 'see'.

Element1 .Tel

- Element level module, for containing Element level structures, behaviours, requirements. Can 'see' the SKAO Library and its own Element specific Library, and everything they can see.

SKAO Library

- Contains common definitions for use across Elements e.g. interfaces, L1 requirements, actors, flows. Can see only the SKAO Profile.

Element 1 Library

- Contains elements that are common across Element1 .Mid and Element1 .Low. Can see the SKAO Library and the Element 1 Profile, and everything they can see.

SKAO Profile

- Contains customised element definitions (stereotypes) for use across the model.

Element 1 Profile

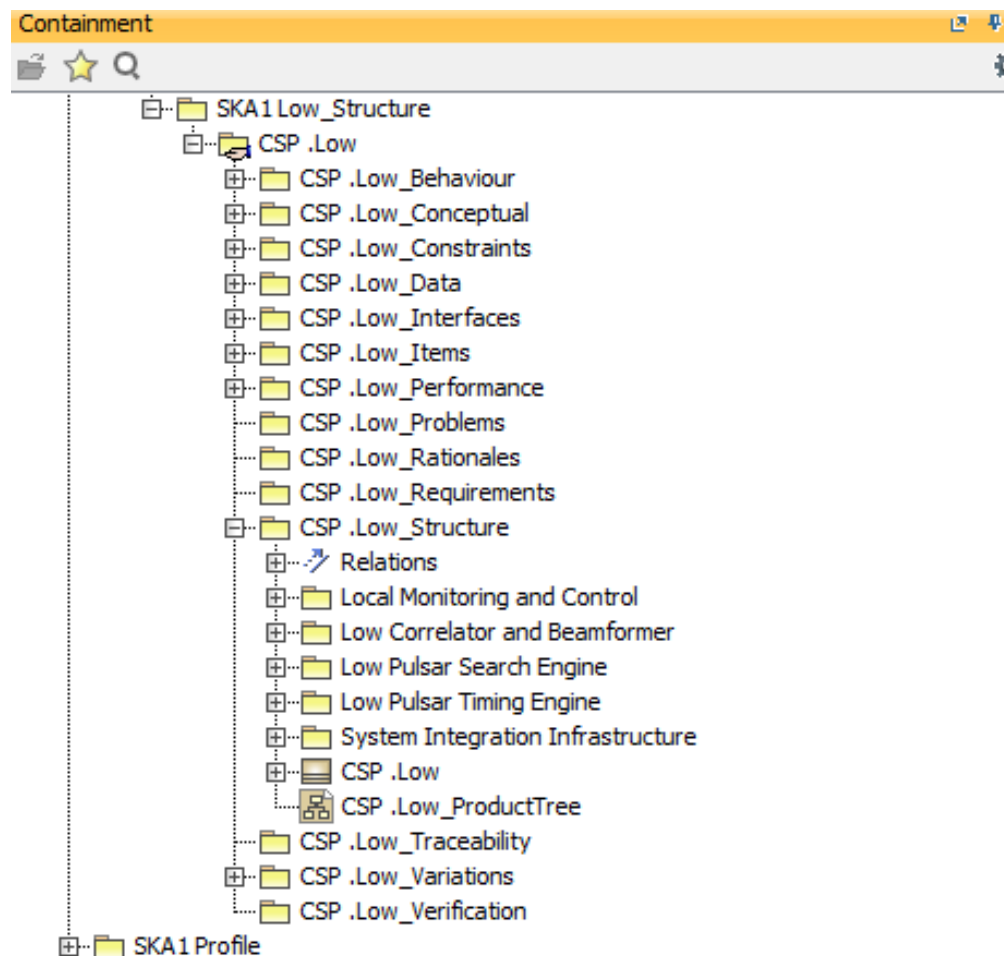
- Contains customised element definitions (stereotypes) for use only in Element 1 modules.

# Some implications of this structure

- The **SKAO Library** contains common interfaces and L1 requirements. Because each Element level module can see this library, they can use these interfaces and requirements but the usage is consistent with the common definition
- The **SKA1 Tel** modules can ‘pull up’ content from all the Element level modules they can see, and connect them up to make a ‘working telescope’.  
**But.** Working ‘top down’ across the Telescope / Element boundary is more cumbersome

# SE2 framework

- One-click mechanism to create **standardized package set, stereotyped diagrams and elements**
- Aids consistency, but not perfectly tailored for SKA use
- Options:
  - Abolish usage
  - Use sub-set of features (**current**)
  - Create new tailored SKA framework. Not simple





# SKA consortia model - advantages



“In my experience I have found that as the design becomes more detailed the information becomes too much to use and too wide to remain as a whole in the mind of an engineer. Each time you proceed to a next layer of breakdown the entities increase exponentially. So the further along you have progressed, the longer the “paths” are that must be travelled between cause and effect. ...To have a database that knows the relations between items can be useful to assist a person in understanding the situation.”

– G. Le Roux

# SKA consortia model - advantages

- The tool and language features must be adapted to the SE activities, not the other way around
- If these SE activities are being done across the system, and can be made to be sufficiently consistent, then the model is an **enabler** for doing these activities in an integrated environment where there is a single source of truth



# SKA consortia model - costs

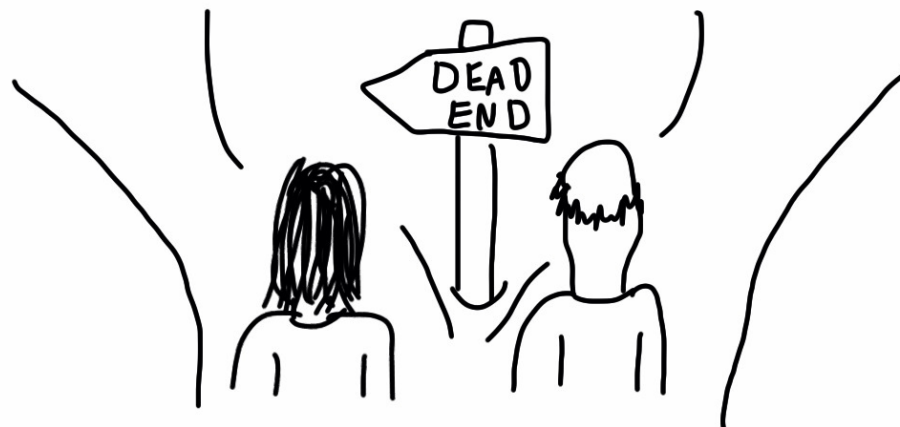
- These are **pragmatic** and include:
  - Resources (cost of modelling resources, training, time spent modelling, learning curve)
  - Limitations of the modelling tool, including problems of inter-operability
  - Configuration managing the model content
  - Achieving consistency across the model effort (of method, skills, etc.)

# Subsequent presentations

- Will elaborate on the different types of modelling that have been / can be done in the context of the SKA consortia model

According to the sign  
we should go right

But our model  
says left



[freshspectrum.com](http://freshspectrum.com)

Thank you