Overall System Modelling

OSMoSE

For Space System Engineering

MBSE-2021 – Space System Ontology Workshop

Conceptualizing MBSE – Part 1

MB4SE

model based for system engineering

Elena Alaña, Jorge Pacios

ESA ESTEC

30/09/2021

The Space System Ontology Development project

Project:

- Space System Ontology Development (SSOD-GMV)
- 12 months activity (KOM: 29th October 2020)

Scope:

1. UoD Functional Description static part
2. UoD Architecture/Logical Description
3. UoD Physical Description

Consortium:

- ESA – Serge Valera (Technical Officer) and Quirien Wijnands
- GMV – Main Contractor
- Airbus DS, Thales Alenia Space, OHB - Subcontractors

Main concepts to address

1. UoD Functional Description static part
    - Function
    - Function tree, functional categorisation, functional architecture
    - Function port, functional exchange, functional chain
    - Component, component actor, component port, component exchange
    - System, system function, system functional exchange, system functional chain, system actor
2. UoD Architecture/Logical Description
    - Logical function, logical functional exchange, logical functional chain
    - Logical component, logical actor
3. UoD Physical Description
    - Physical function, physical function exchange, physical functional chain
    - Hosting component, physical port, physical link

Conceptualisation

- Language and tool:
  - ORM – Object Role Modelling
  - NORMA Pro
  - *The Reverse Engineering of the Capella model is used as starting point* *(removing implementation specifics and injecting semantics)*

- Modelling Approach:
  - Iterative process – every stakeholder viewpoint is injected in the model
  - “Global as a Whole” and “Locals as Views”

- Drivers:
  - The utilisation of the Ontology must govern its development
  - Based on use cases (real interoperability scenarios)
  - In line with ECSS-E-ST-10C + Change Requests (issued as needed)

Process

1. Inputs gathering:
    - Identification of what is exchanged without considering the names, e.g. *data produced for a specific milestone*
    - Real examples are used, e.g. *Function Tree*
2. Assessment:
    - Examples presented to the consortium in weekly Working Sessions
    - Identification of how other companies represent the same concepts
    - Agreement of the best conceptualisation approach considering all views
3. Conceptualisation in the ORM model
4. Review of the ORM specification (NORMA) and the requirements specification (Word/PDF)
    - Internal review to confirm that the final conceptualisation is acceptable to everybody
    - Review by the OSMoSE Governance group + MB4SE Advisory group (*not started*)
    - Review and acceptance by the OSMoSE Design Authority group (*not started*)

Main challenges

- Focus on WHAT is exchanged
  - Address the level of detail needed by the exchanges
    - Information is not always generated from tools, and Excel, Visio,… are used instead
    - The real engineering information/artefacts defined in a document are not obvious
  - Clarify and detail what is exchanged
    - Exchanges informally defined today (e.g. communication means without formal semantics)
    - Terms are not used in the same way by all partners involved
    - Same terms used in a different way
  - Avoid being biased by any MBSE methodology/tool, *e.g. All information needed by an authoring tool is not exchanged*
- Reach consensus
  - The Ontology shall represent all needs *(all communities shall be represented)*
  - The same artefact exchanged is not always represented in the same way *(relations and naming)*
  - The alternatives shall be analysed to identify commonalities and variabilities
  - All the elements *(Object types, fact types, constraints)* conceptualised in the Ontology shall be agreed by all parties

The Static View of the Functional Description

Here are compiled some examples for
the Static View of the Functional Description UoD

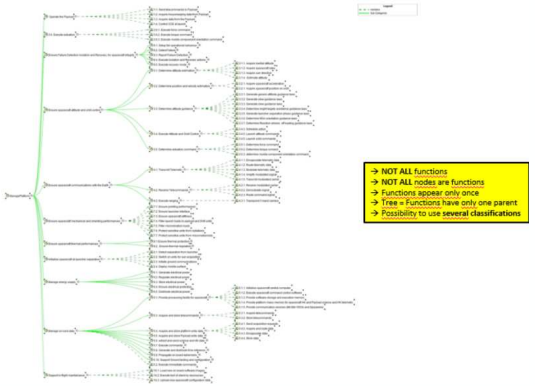
Function Tree

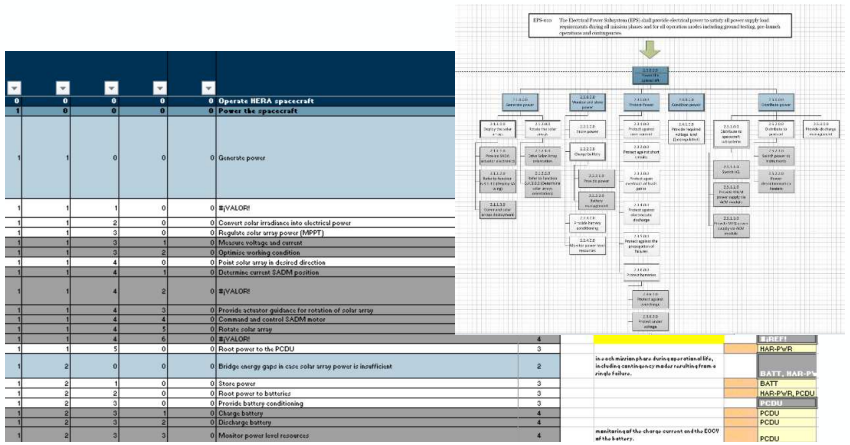
ECSS-E-ST-10 requires exchange of Function Trees but each LSI interprets differently what a function tree is !

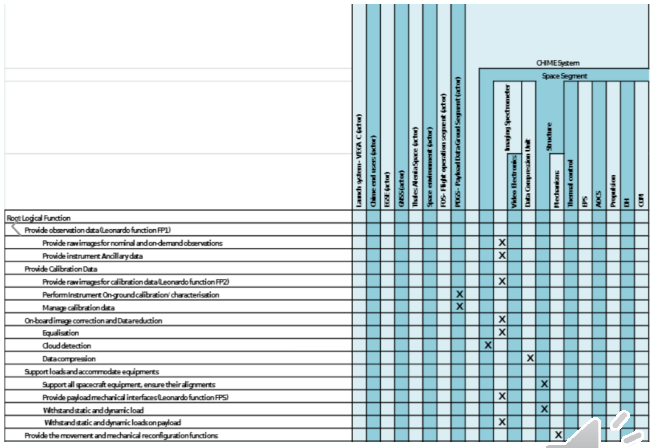
Definition: *hierarchical breakdown of a function into successive levels of function*

Function Trees examples produced by Thales, Airbus and OHB were analysed

- They are compliant with the ECSS definition
- Their semantics differ !



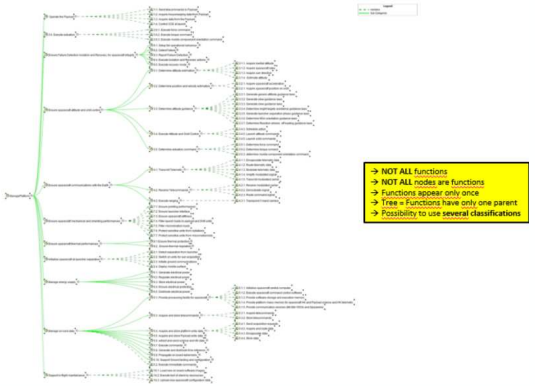


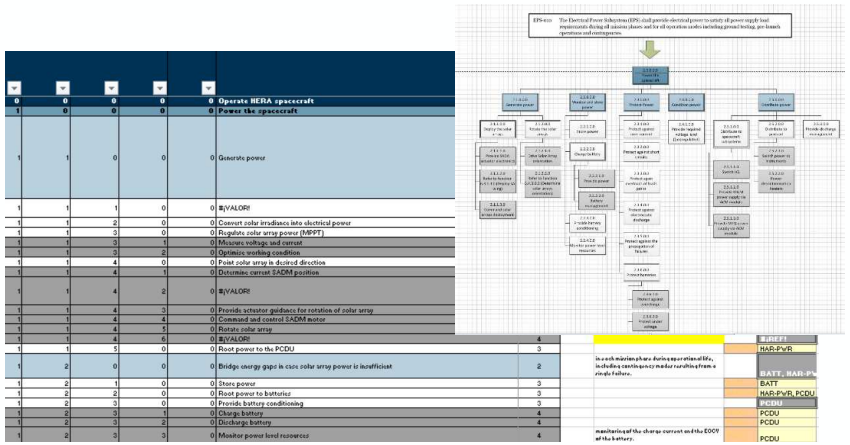


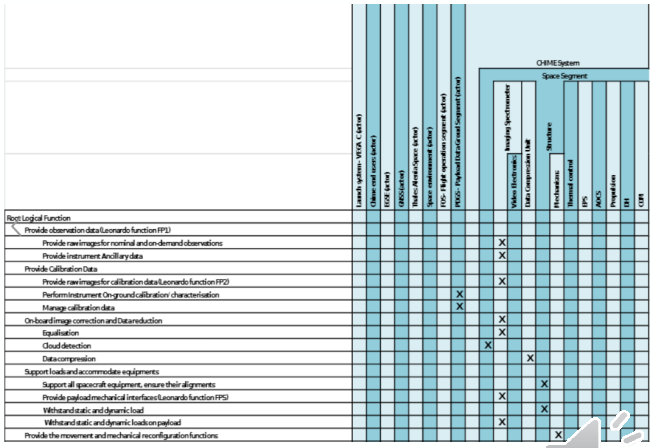
Function Tree

Examples of differences:

- Definition of a single Function Tree vs. several Function Trees
- Functions repeated in different nodes (meaning that the same function contributes to the parent function) vs functions not repeated
- Only functions are included vs. *categories* can be also defined
- …







Function Tree

All stakeholders have agreed on a **specification** of the Function Tree **concept** (expressed in ORM), compliant with the E-10 definition:

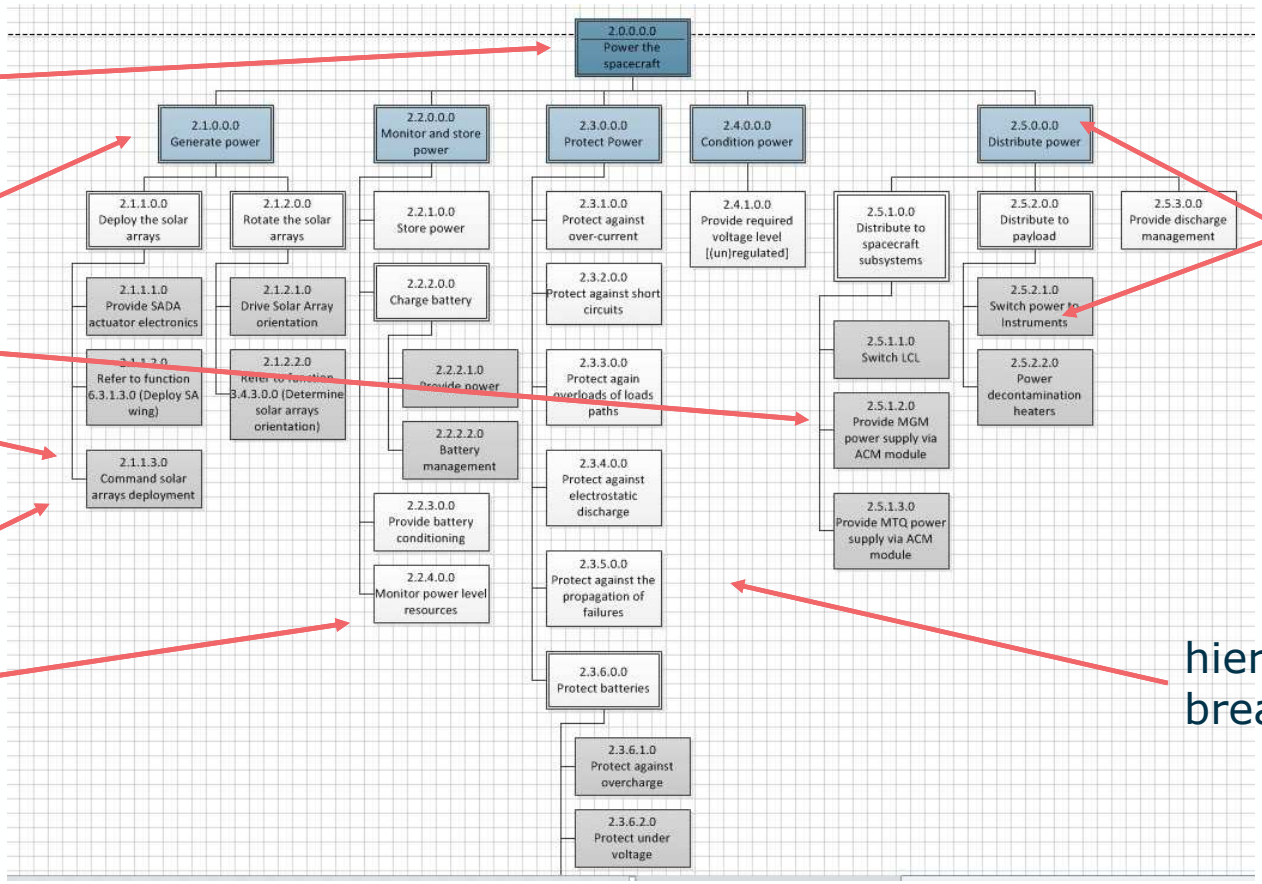
- all function tree nodes are functions;
- functions can be reused, e.g. if a function is performed by an off-the-shelf product;
- exchanged function trees must be fully defined, i.e. all functional nodes must be exchanged.

In order to specify what a Function Tree is a definition is not enough.

- what are its constituent parts?
- what are the relationships with other concepts?

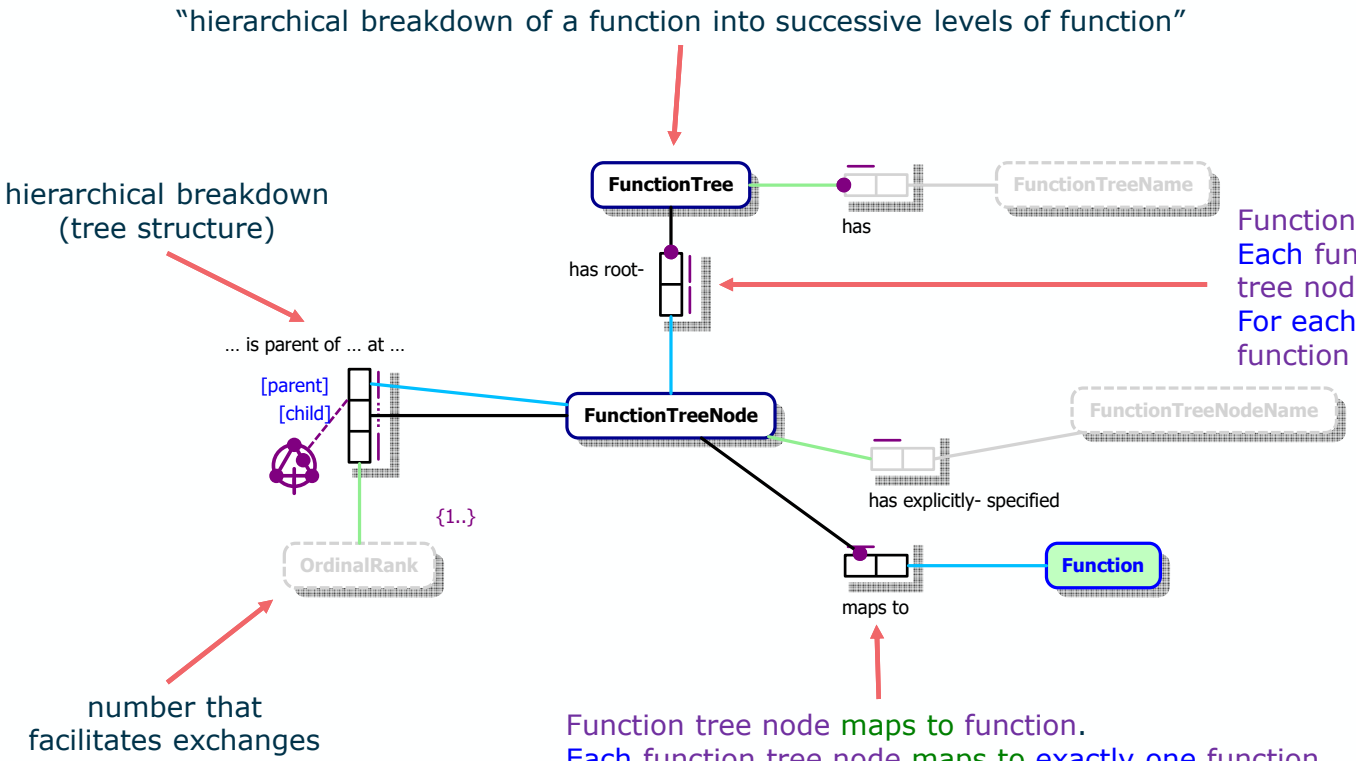
Function Tree

Example



Function Tree

Model: essential fact types



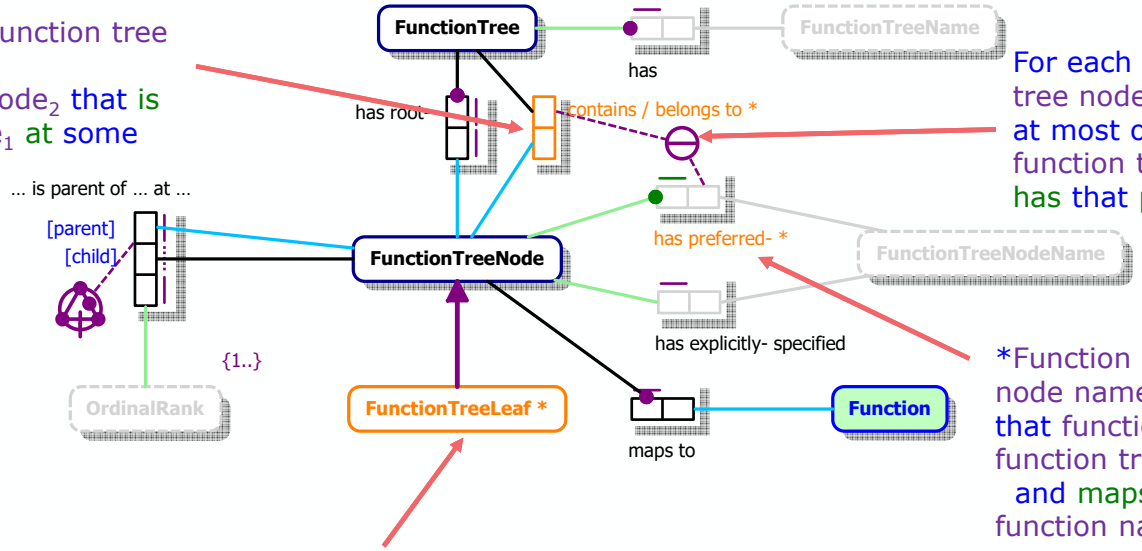
Function tree has root function tree node.
Each function tree has exactly one root function tree node.
For each root function tree node, at most one function tree has that root function tree node.

Function tree node maps to function.
Each function tree node maps to exactly one function.
It is possible that more than one function tree node maps to the same function.

Function Tree

Model: additional fact types, derivations and constraints to ensure full consistency of the population

*Function tree contains function tree node$_1$ if and only if
that function tree has that root function tree node$_1$
or contains some function tree node$_2$ that is parent of that function tree node$_1$ at some ordinal rank.



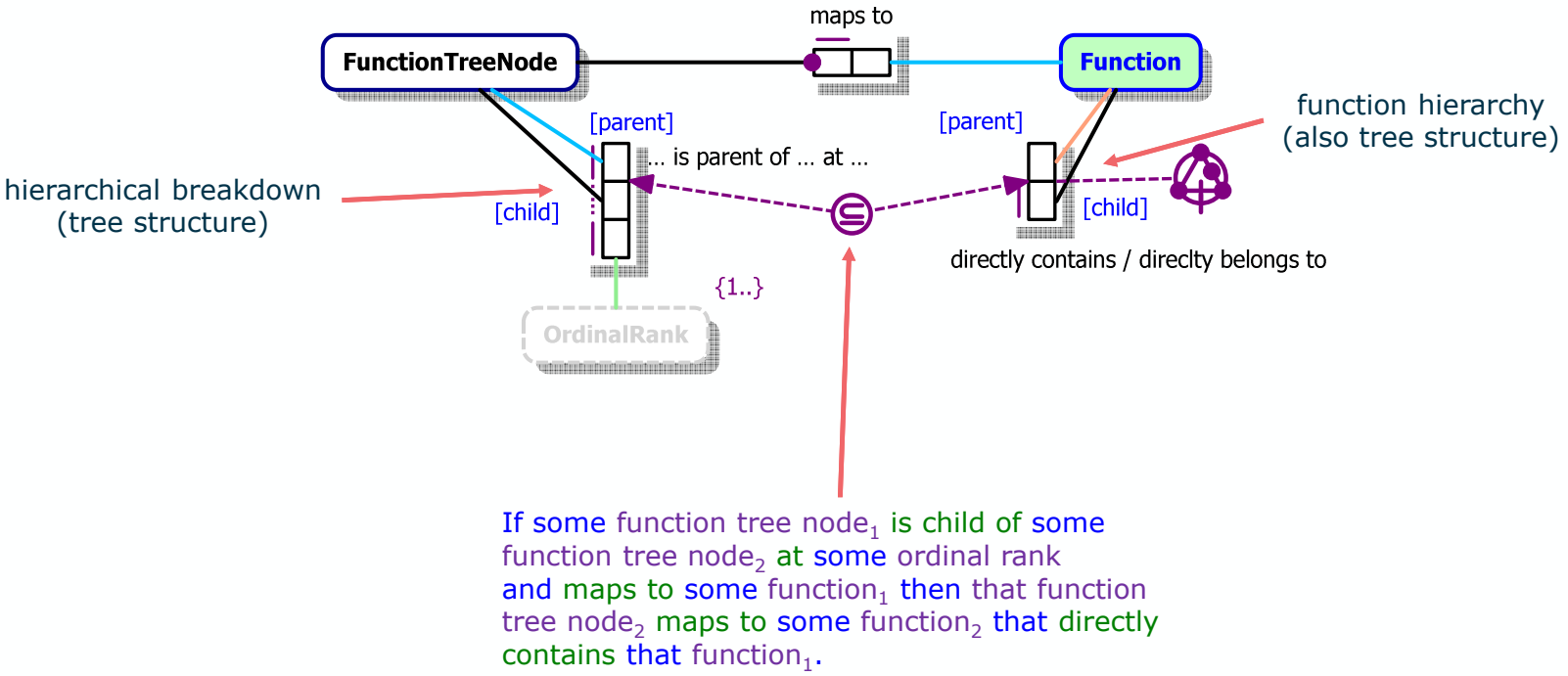
For each function tree and preferred function tree node name,
at most one function tree node belongs to that function tree and
has that preferred function tree node name.

*Function tree node has preferred function tree node name$_1$ if and only if
that function tree node has no explicitly specified function tree node name
and maps to some function that has some function name that is that function tree node name$_1$
or that function tree node has that explicitly specified function tree node name$_1$.

*Each function tree leaf is by definition some function tree node$_1$ where it is not true that (that function tree node$_1$ is parent of some function tree node at some ordinal rank).

Function Tree

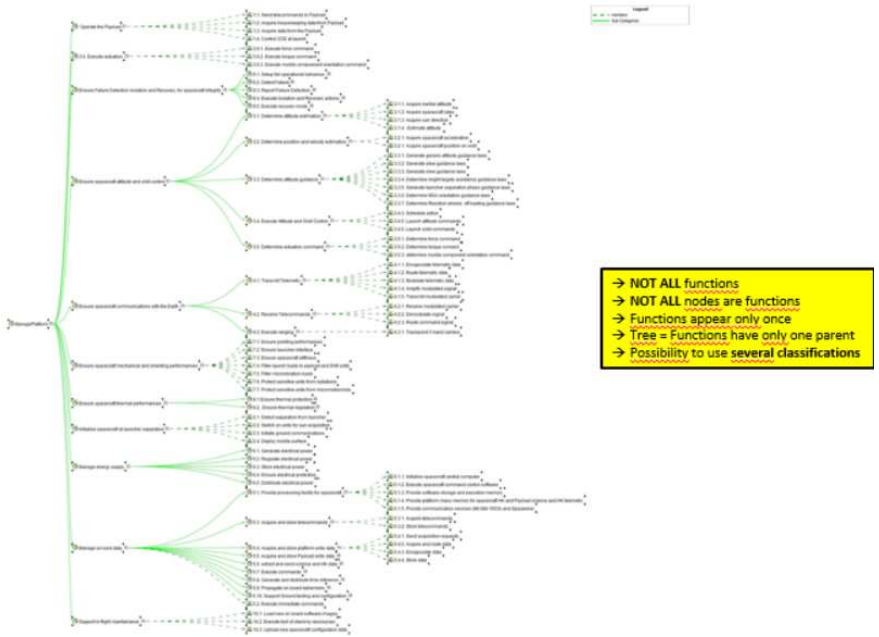
Function tree vs. function hierarchy: the function tree can be a subset of the global function hierarchy (we are not forced to share from the root), but is driven by it.



Functional Categorisation Another viewpoint

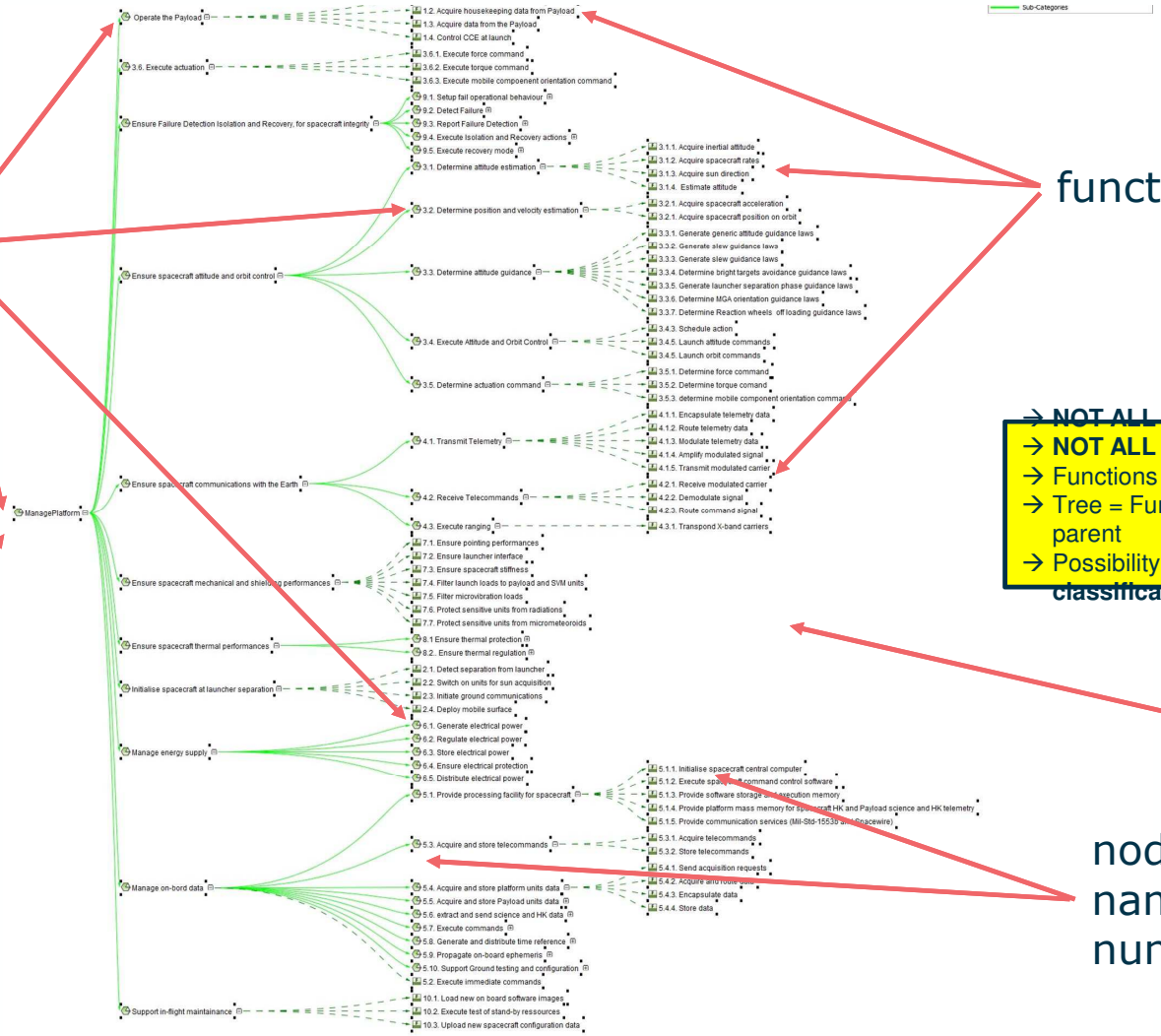
OSMoSE differentiates between the concept of Function Tree and the concept of Functional Categorisation

- Discovered while discussing the Function Tree concept.
- Exchanging taxonomies of functions is considered convenient by the primes.



Functional Categorisation

Example



functional categories

root functional category

functions

- → **NOT ALL** functions
- → **NOT ALL** nodes are functions
- → Functions appear only once
- → Tree = Functions have only one parent
- → Possibility to use **several classifications**

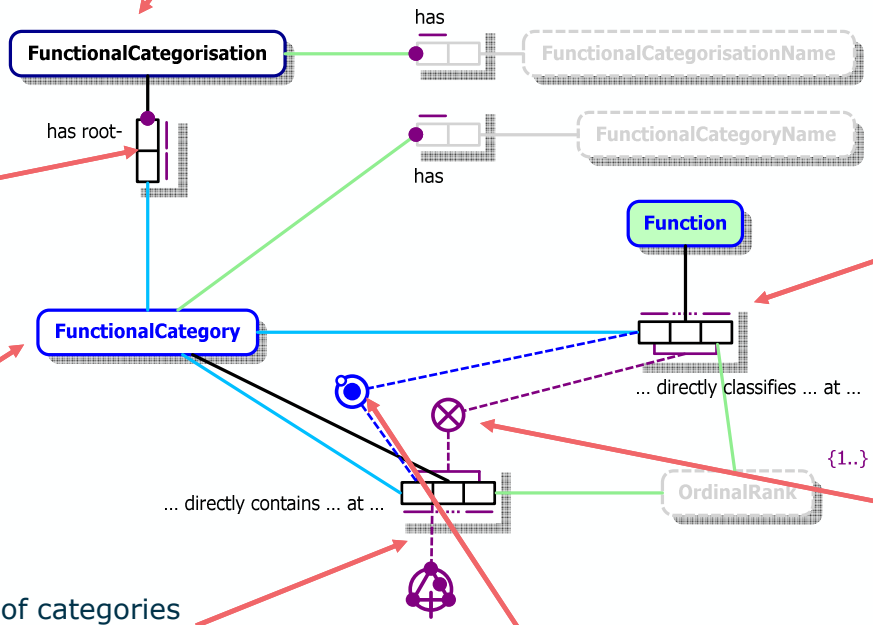
hierarchy represents classification

nodes have names and numbers

Functional Categorisation

Model: essential fact types

classification of function into successive levels of functions or functional categories



Each functional categorisation has exactly one root functional category.
For each root functional category, at most one functional categorisation has that root functional category.

functions are classified by categories, with an ordinal rank to improve the exchange

common attribute to classify functions

For each functional category and ordinal rank, at most one of the following holds:
that functional category directly classifies some function at that ordinal rank;
that functional category directly contains some functional category at that ordinal rank.

hierarchy of categories (tree structure)

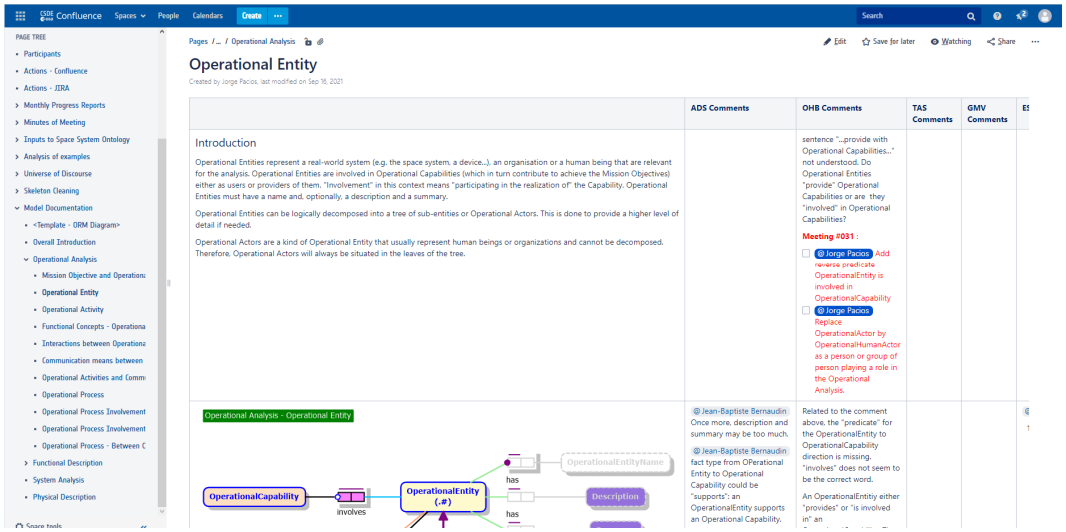
It is obligatory that each functional category directly contains some functional category at some ordinal rank
or directly classifies some function at some ordinal rank.

Conclusions

- All stakeholders can keep on representing their Function Trees in the same way using their authoring tools.
- Interoperability is enabled since all semantics are formally expressed.
- Concepts apparently clear hid semantically different interpretations that are now agreed.
- Ambiguity has been removed. Functional Categorisation could become a change request for ECSS.
- The use of examples is essential. A graphical representation of the examples is often required to achieve a better understanding.

Status

- Documentation is being produced for the UoD Functional Description (static part), to be reviewed by the partners.
- Modelling of the UoDs Architecture/Logical Description and Physical Description is in progress.
- Documentation of Operational Analysis (coming from Arcadia) is under review.
- The group is using ESA's Confluence pages to support discussions, publish documentation and perform the reviews.



Questions?