The Swarm constellation simulator

A brand new, but still operationally responsive development

Torrance, CA

24th March 2009

Max Pignède [*presenter*], Mario Merri, Vemund Reggestad

European Space Agency (ESA) / European Space Operations Centre (ESOC)

Peter Fritzen, Michael Irvine

VEGA Deutschland GmbH & Co. KG / A Finmeccanica Company

A Finmeccanica Company

What this presentation talks about…

1. Why attempting a new approach in making simulator software ?
2. “Operationally responsive” architectures and technologies
    a. SMP2
    b. REFA
    c. EGOS-MF
3. Benefits for the users
4. Conclusions

Rationale -- Background

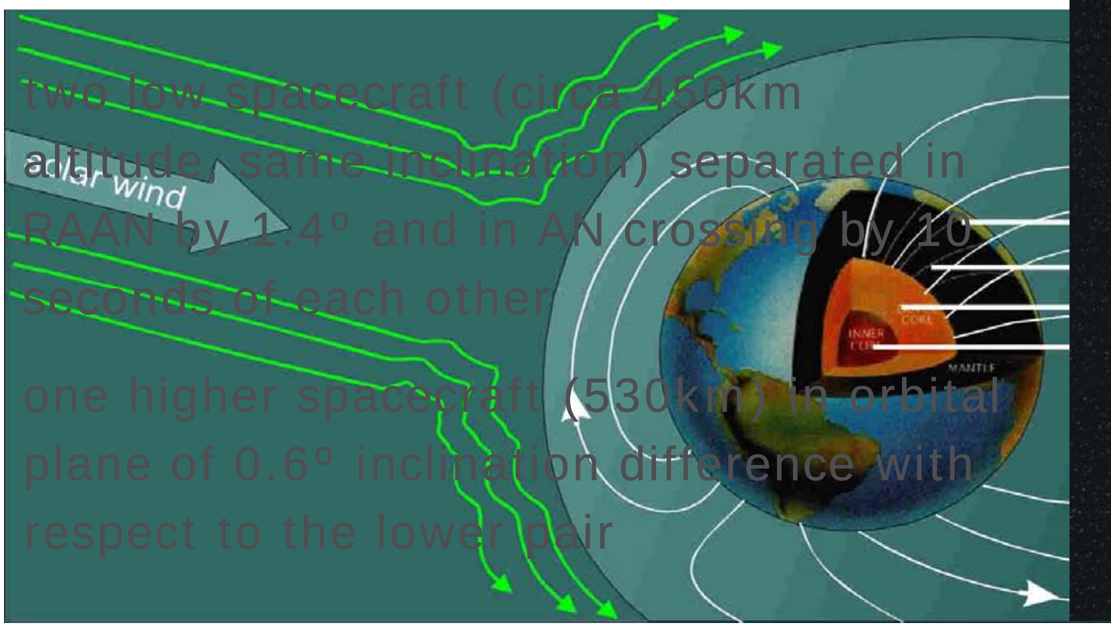
1. Recognition that development strategies better than mere reuse should be explored
2. Take advantage of new technologies and of ESA most recent standards
3. ESA (ESOC) is preparing the first deployment of a new generation of operational simulators in the context of the Swarm mission, a constellation of three satellites

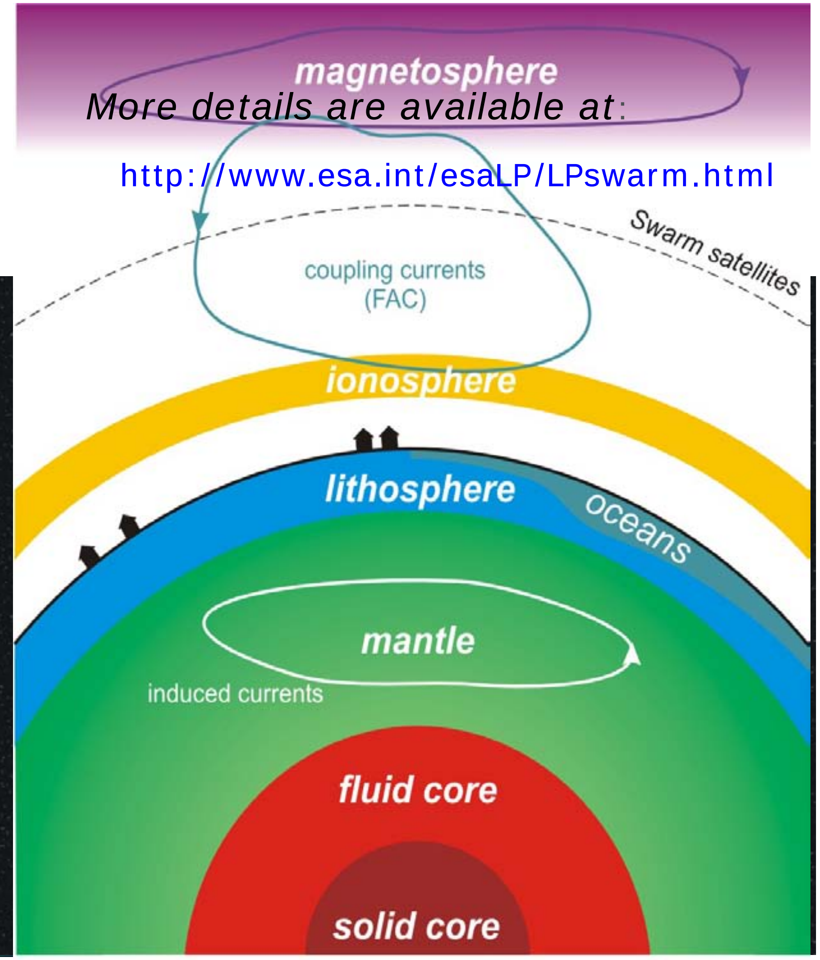
Overview of the Swarm mission

1. Study the dynamics of the Earth magnetic field and its temporal evolution
2. Improve understanding of its various contributing sources (Earth core, Earth mantle and lithosphere, magnetosphere and ionosphere)
3. two low spacecraft (circa 450km altitude, same inclination) separated in RAAN by 1.4° and in AN crossing by 10 seconds of each other
4. one higher spacecraft (530km) in orbital plane of 0.6° inclination difference with respect to the lower pair

More details are available at:

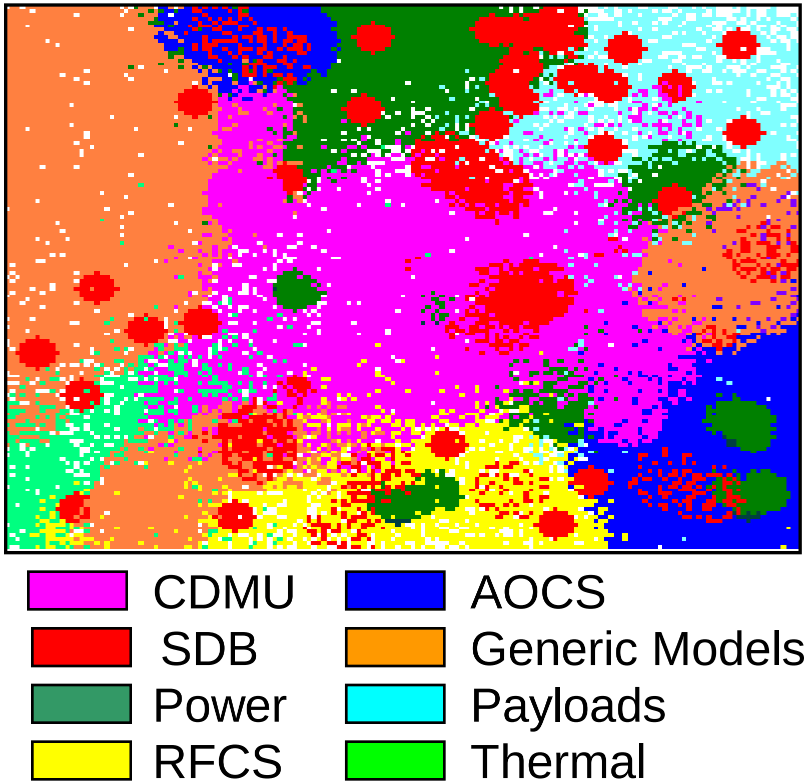
http://www.esa.int/esaLP/LPswarm.html





Why do we need a reference architecture?

Example of what a simulator software architecture may look like:



1. No clear interface between models
2. Difficult to isolate models for reuse
3. Tight coupling of database and models
4. Reuse potential mostly only via “copy&paste”

How about if we …

a. defined clear interfaces between the different elements in a spacecraft ?
b. defined some suitable breakdown of a whole simulator into models ?
c. improved reusability at model level in a “plug&play” fashion ?

By the way … simulators: what is this and why?

1. This is a computer running a satellite related application software
    a. it provides the operator with the same view on the satellite as of the real satellite
    b. what matters for ground operators is essentially TM and TC

2. A simulator includes typically:
    a. an on-board software emulator
    b. functional models of satellites hardware equipments
    c. and … an operator interface

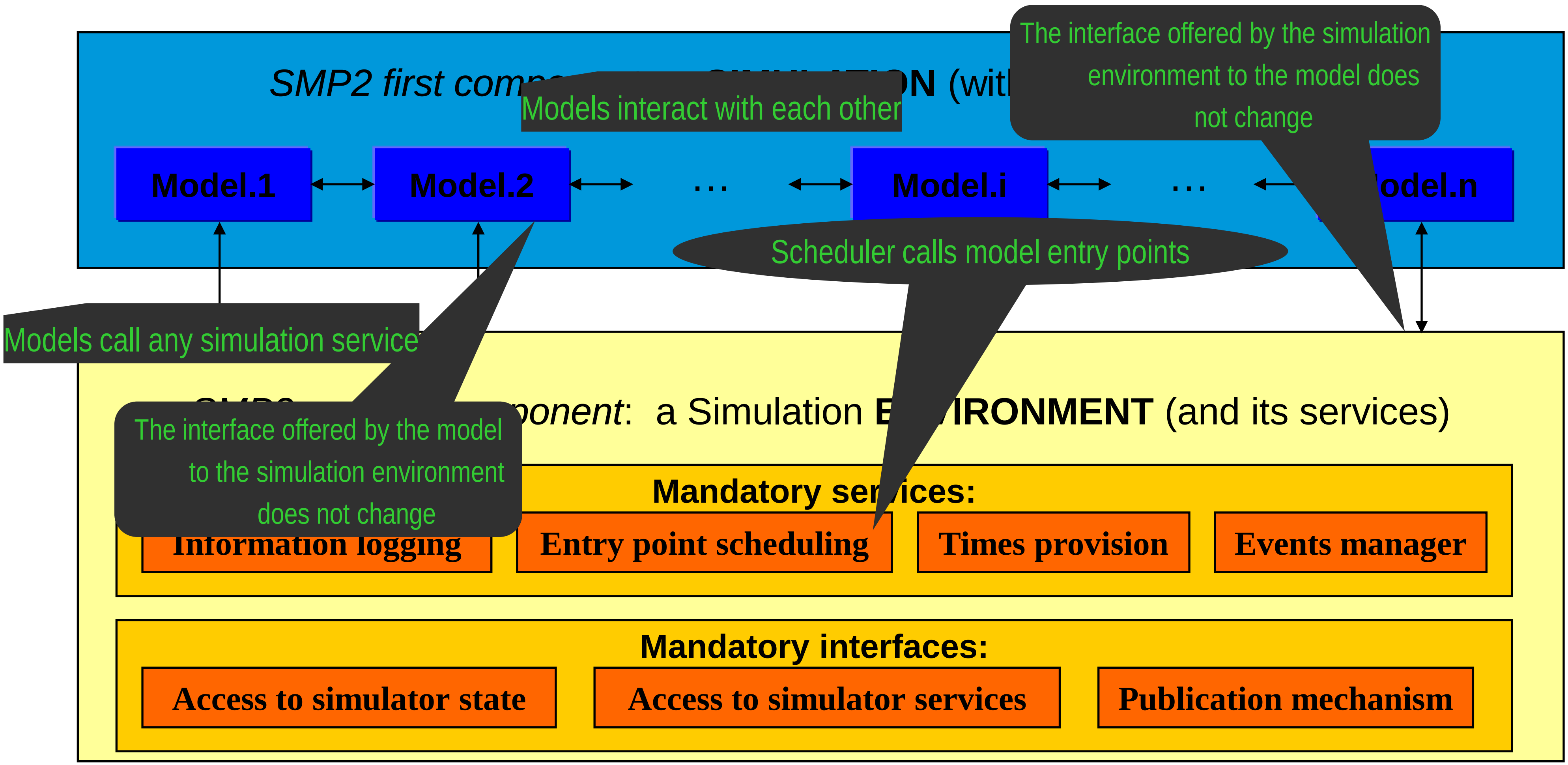
3. Typical simulator usage scenarios are …
    a. mission control system and facilities testing
    b. validation of the operational procedures
    c. training of operations staff -- with e.g. injection of contingencies

“Operationally responsive” technologies

1. The **SMP2** Standard ...
   a. promotes portability of models among different simulation environments and operating systems
   b. promotes the reuse of simulation models
   c. fulfills these objectives by providing a standard interface between the simulation environment and the models

2. The Reference Architecture (**REFA**) ...
   a. identifies, using SMP2, a reference spacecraft simulator architecture which can be used as the basis for simulators design and development
   b. achieves shorter development cycles by reuse relying extensively on a common architecture
   c. fulfills these objectives by specifying interfaces between spacecraft subsystems and by identifying models which can be developed in a generic fashion

SMP2 architecture



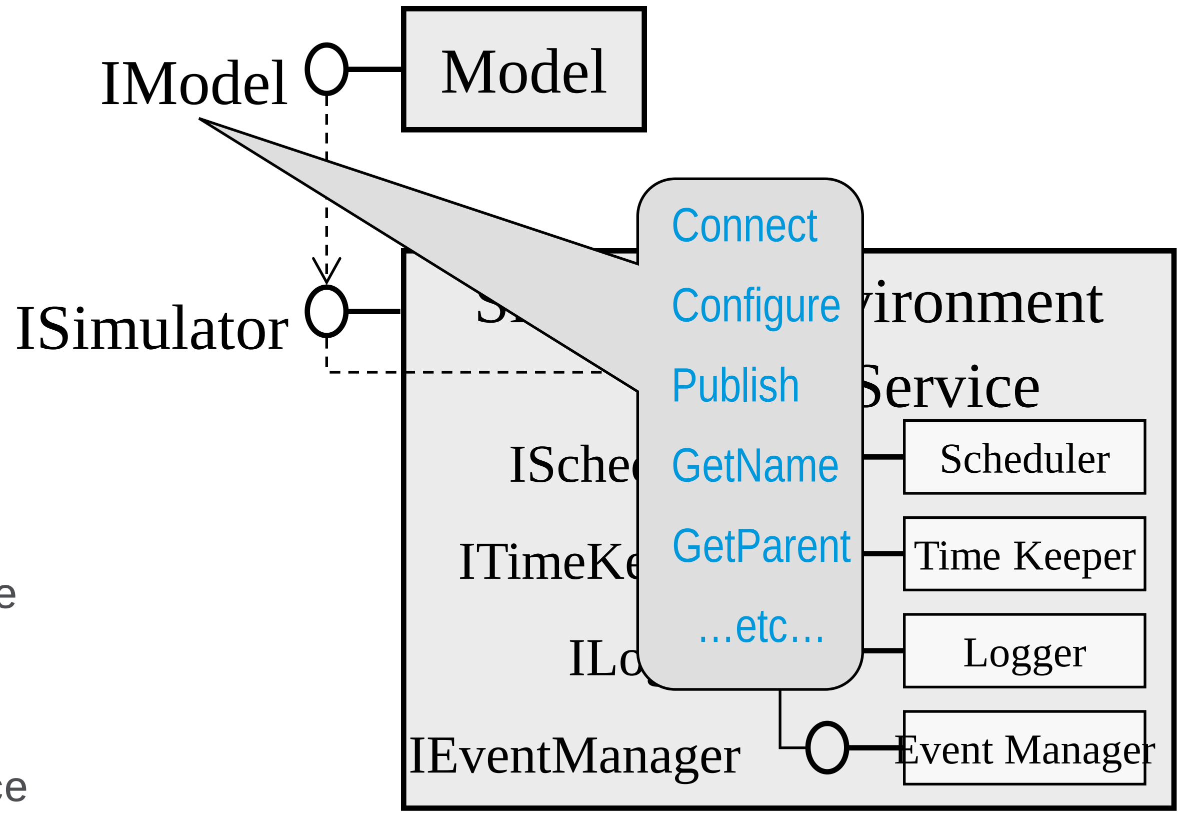
SMP2 simulation services and interfaces

1. Simulation services are …
   by the
   n environment
   he models
   d
   s

Connect
Run
Hold
Restore
GetState
GetTimeKeepe
r
GetModel
…etc…

2. …must implement the …ace

3. Every simulation service must implement the **IService** interface



SMP2 key benefits

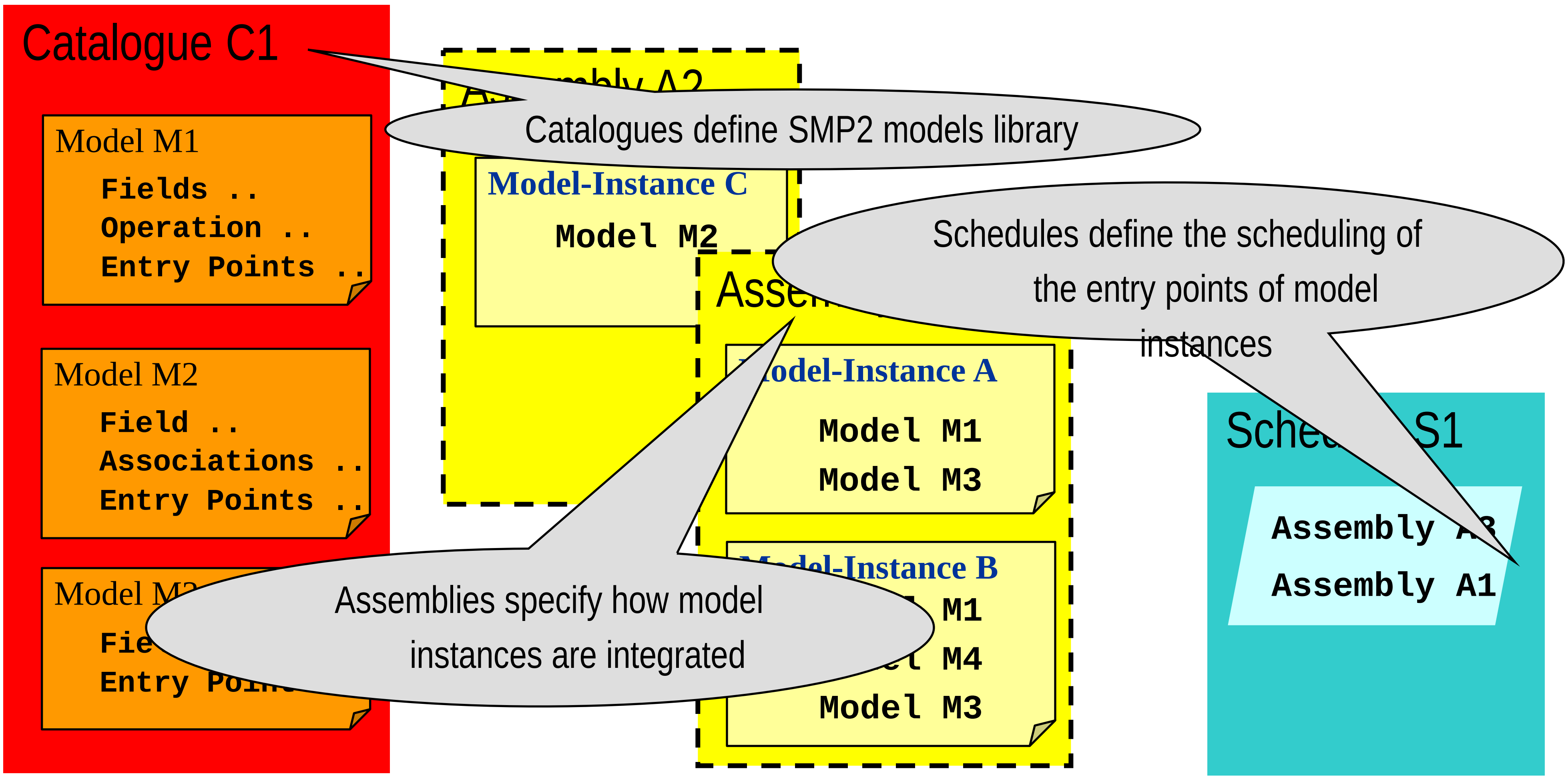
1. SMP2 avoids developing models which use the operating system or hardware specific dependencies -- Platform Independent Model (PIM)

2. SMP2 promotes the use of modern software engineering techniques -- in particular component based design

3. SMP2 makes reuse easier via breaking dependencies between simulator models

4. SMP2 allows dynamic configuration -- for example the user may at "runtime":
   a. select different orbit propagators depending on the required accuracy
   b. switch a component that simulates a hardware equipment with a component that interfaces with the real equipment (when moving from simulated models to real equipments)

SMP2 hierarchy -- the main parts



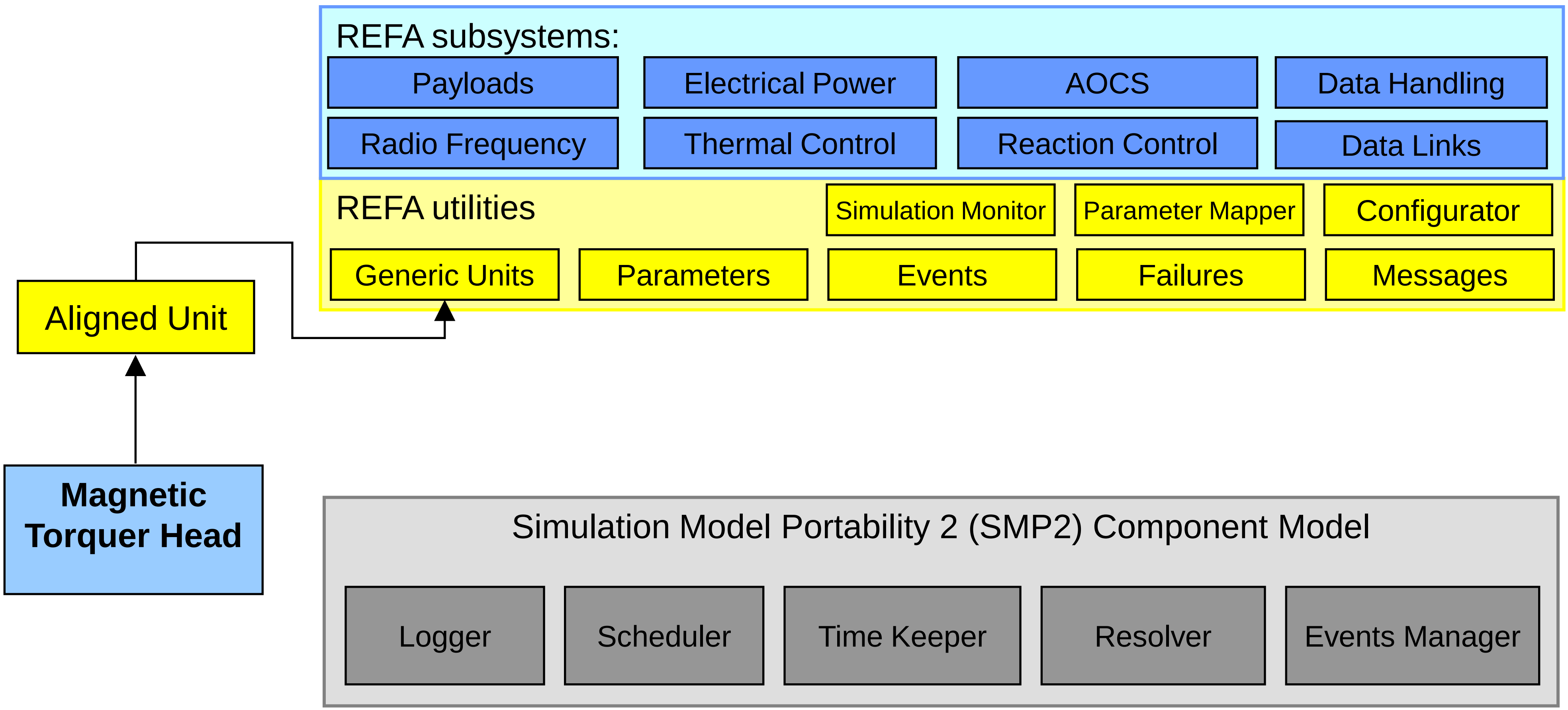
REFA onset

1. REFA started with investigating what would be worth being standardised for all satellite subsystems across simulators

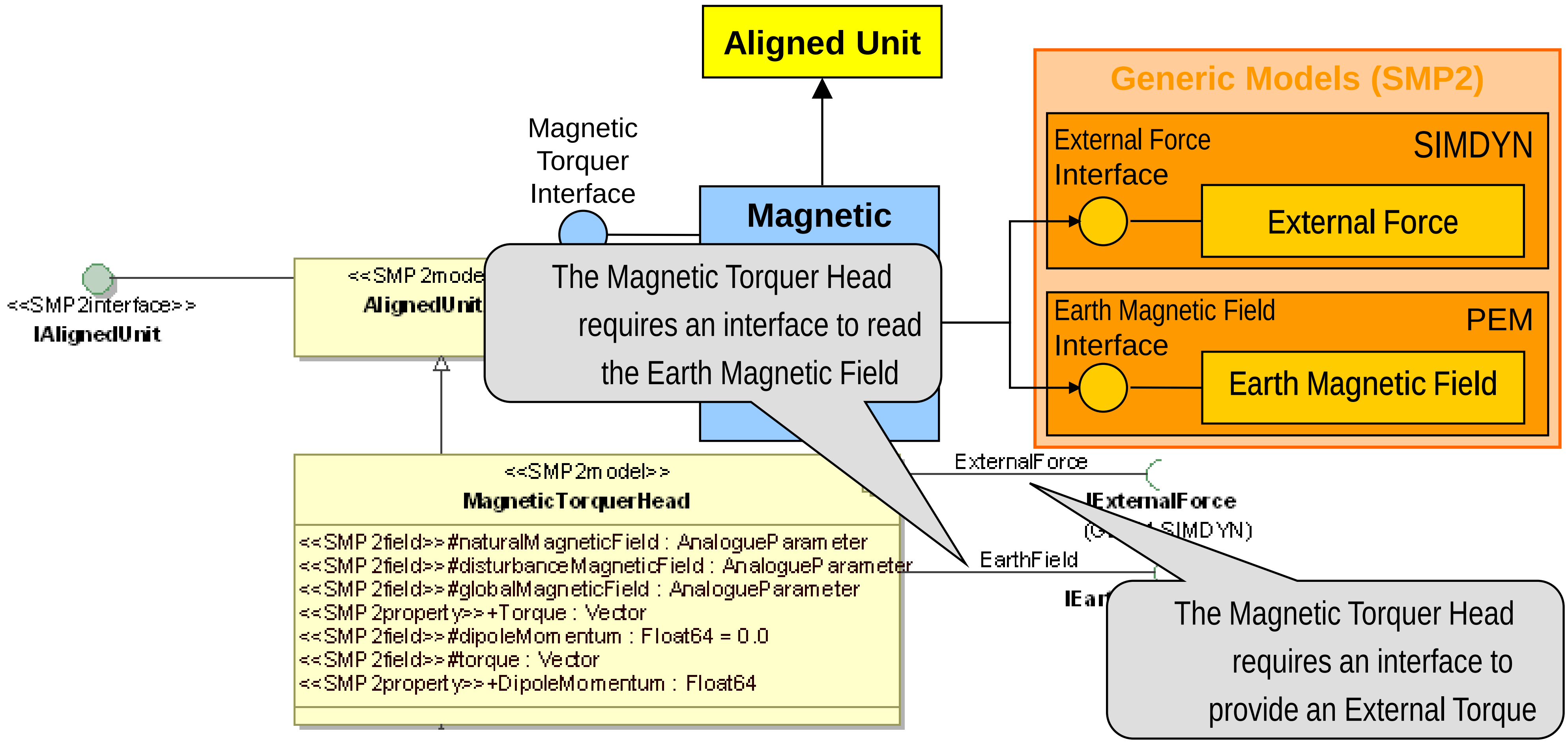
   -- via the actual screening of 4 different space missions

2. As a result REFA identified ...
   a. what should the “reference spacecraft simulator” requirements be
   b. all interfaces between the different elements in a spacecraft simulator
   c. the models which can be developed
      – in a generic fashion (e.g. satellite dynamics, orbital environment modelling, satellite thermal control, communications subsystem, ...)
      – those other models which need to be developed specifically for each mission
3. This is a system **architecture** -- some hard work remains to be done ...
4. Use this architecture as the basis for future simulators development !

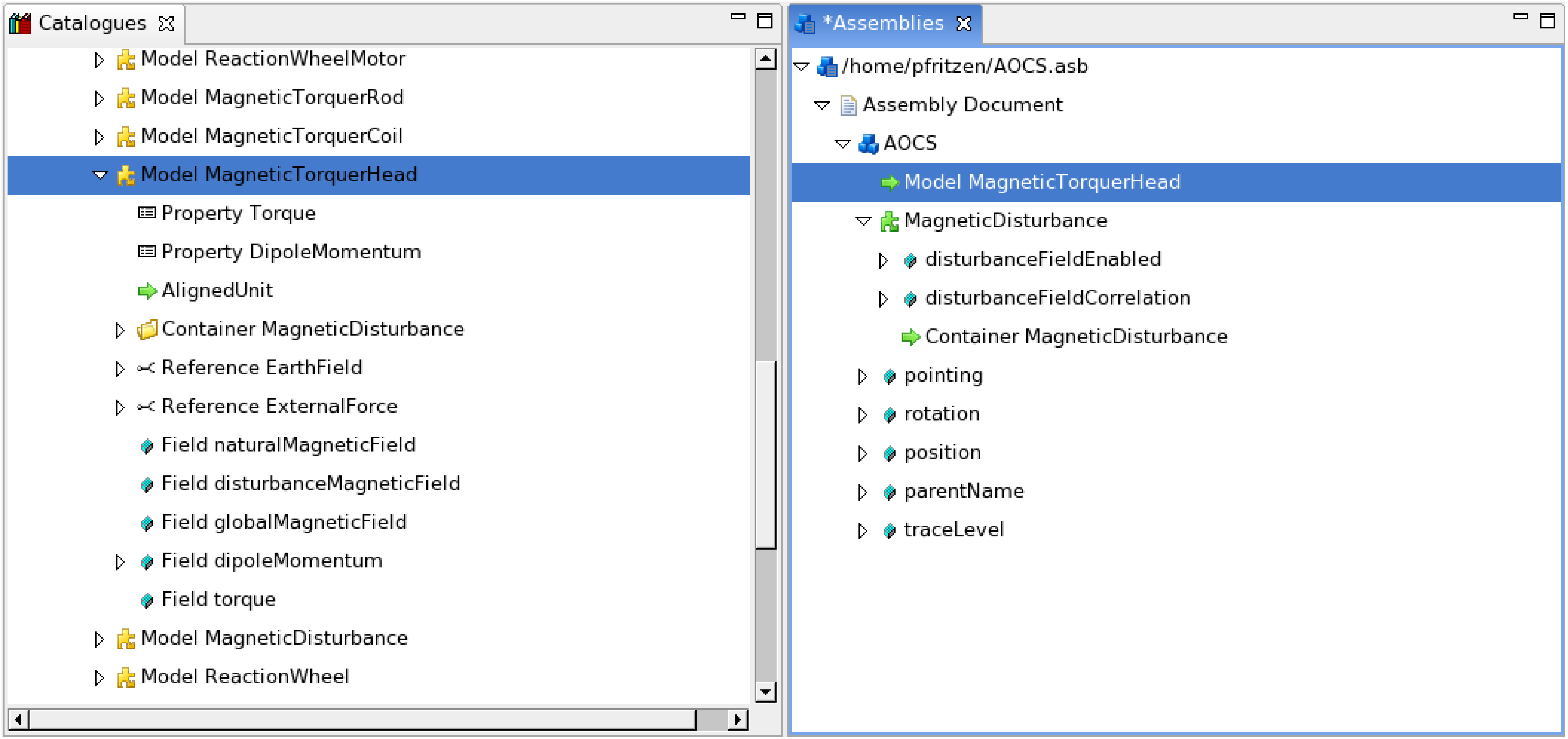
REFA logical model



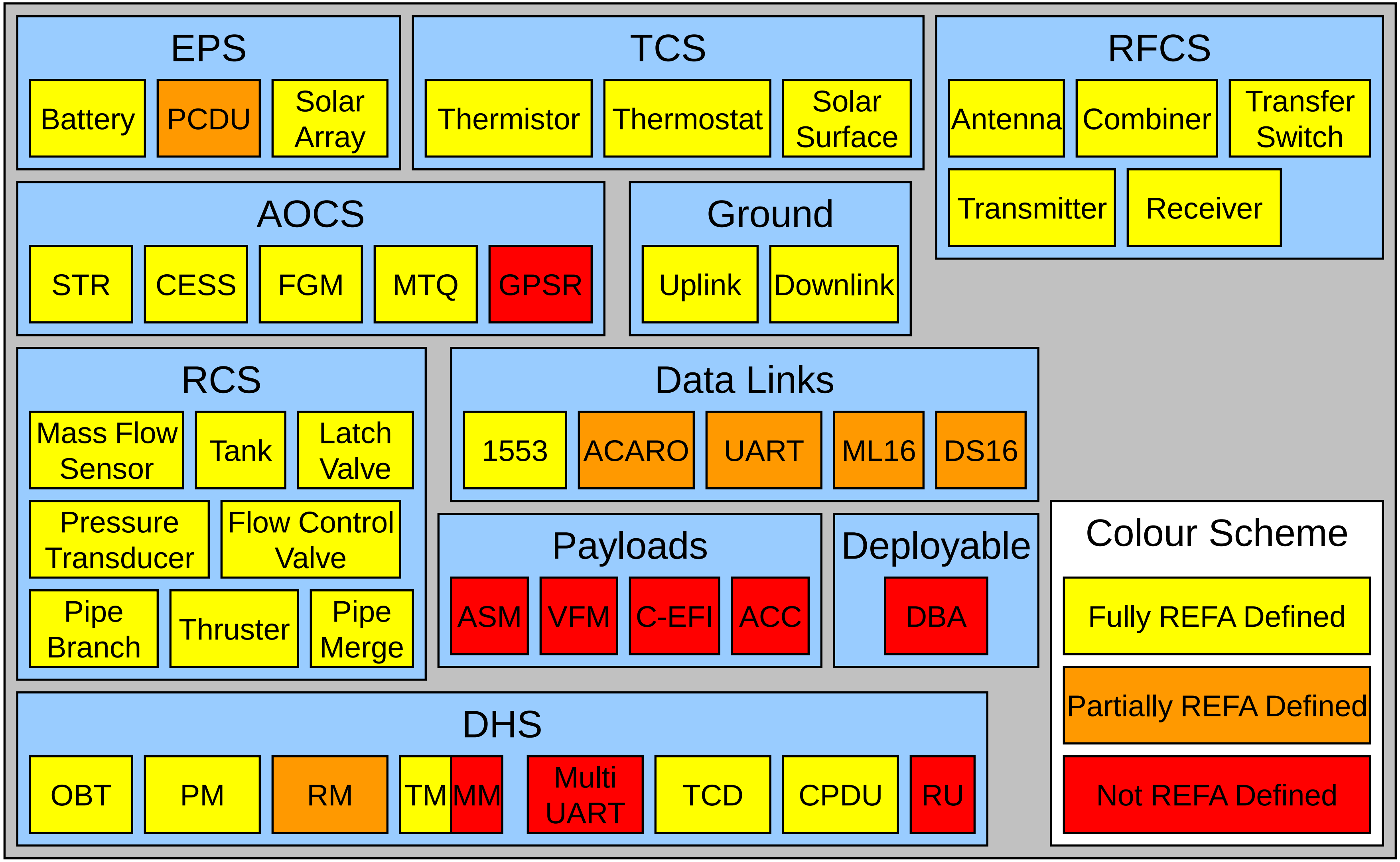
REFA outputs -- example



REFA outputs -- and SMP2 editor support



REFA for building a new simulator ...

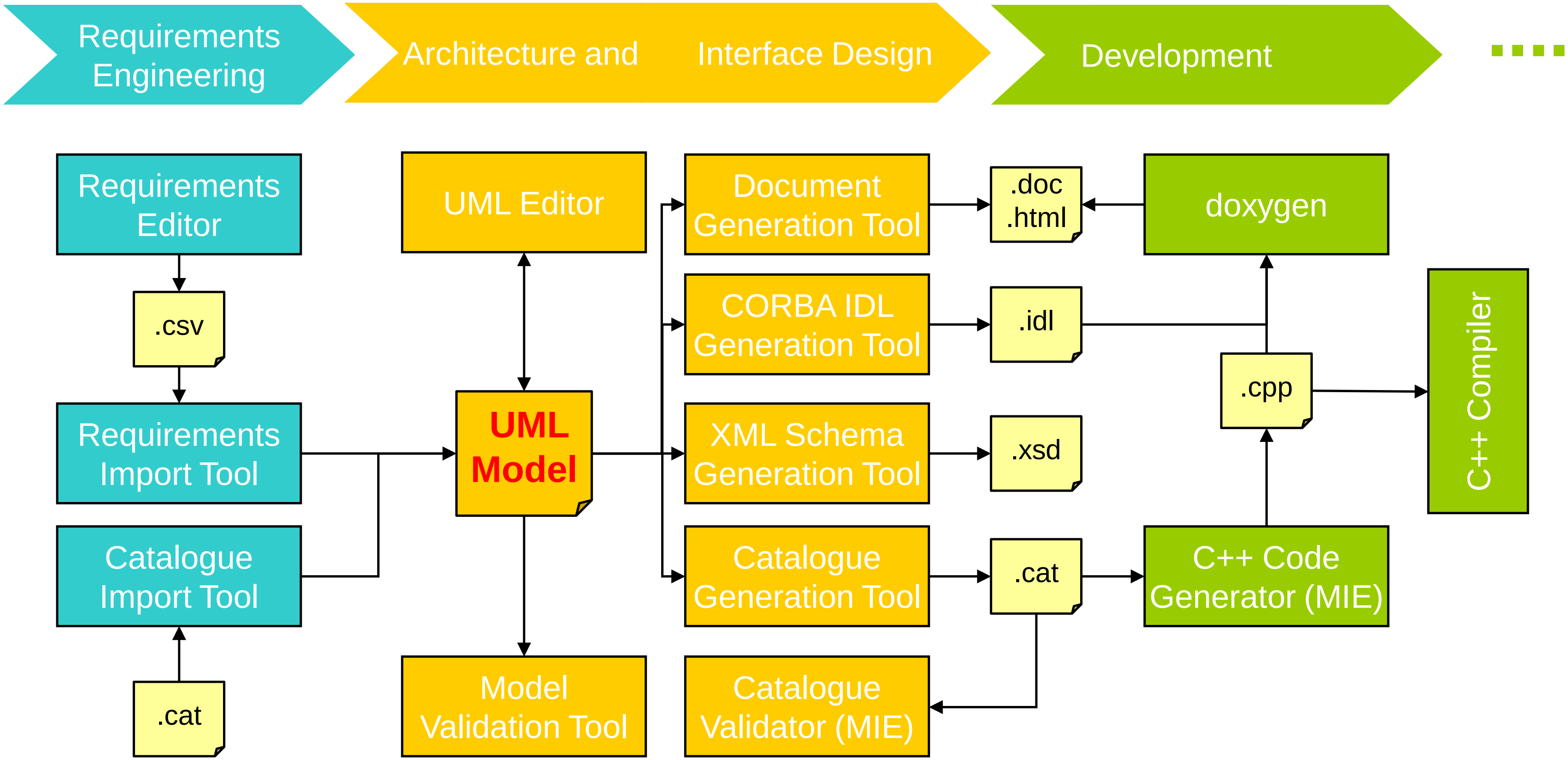


EGOS modelling framework (EGOS-MF)

1. Eclipse based IDE extension for developing SMP2 simulators and for increasing software development efficiency
   a. EGOS-MF is a collection of Eclipse Plug-ins. It uses MagicDraw.
2. Supports the full life cycle of simulator development
   a. design of models -- systematically (re-)start at the UML design stage !
   b. code generation by SIMSAT MIE (including code merge)
   c. document generation (with customisation of generation templates)
   d. model execution and debugging
3. The answer for managing simulators complexity efficiently is ...
   a. to model ! -- and the model is the single source of information
   b. to automate !
   c. and not to repeat yourself !

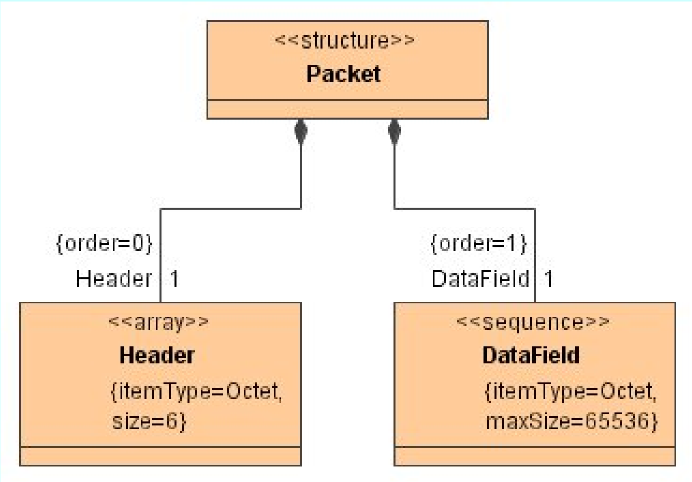
4. Hence: Model Driven Software Development (MDSD) supported by the EGOS-MF suite

EGOS-MF overall development workflow

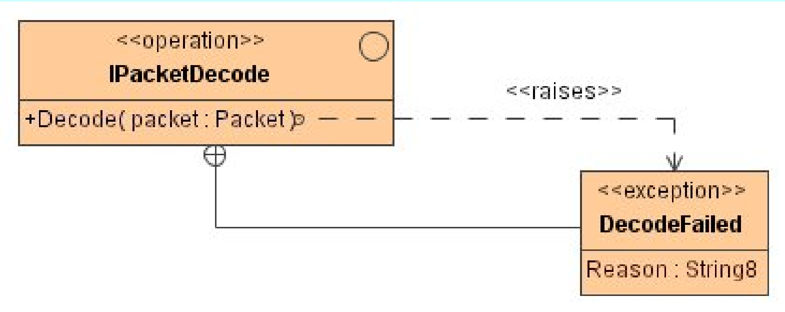


EGOS-MF model driven generation simple case

- UML: **Information Model**



- UML: **Functional Model**



- XML Schema

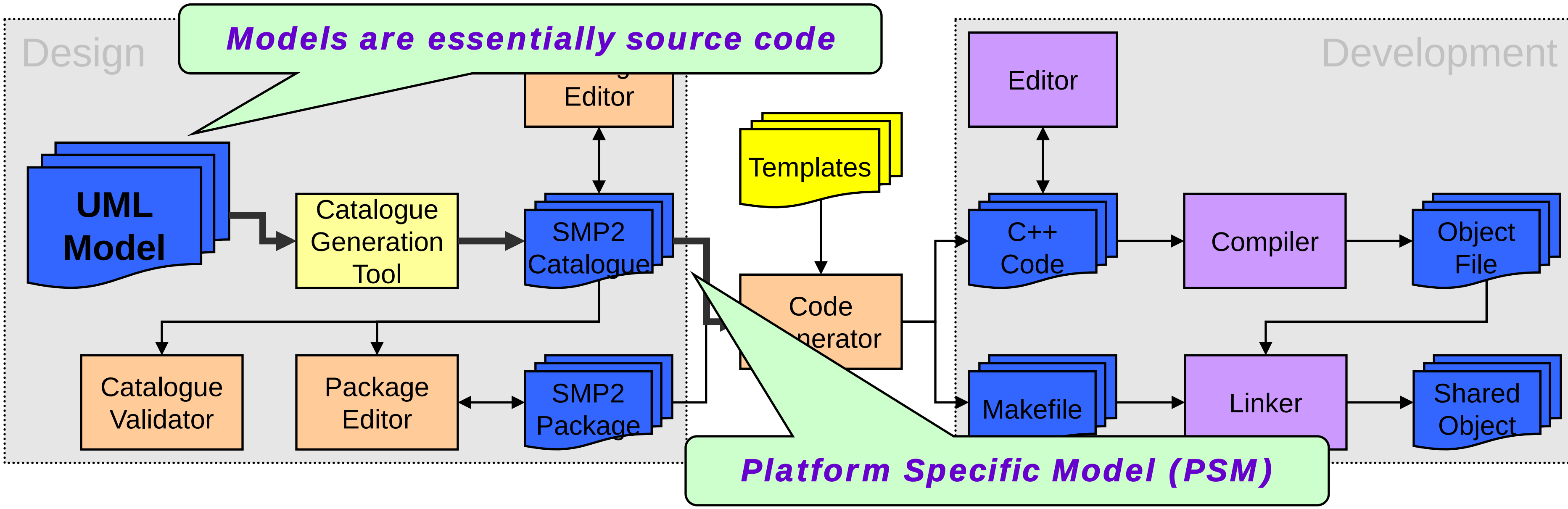
```xml
<!-- ========================================= -->
<!-- <<structure>> Classes                     -->
<!-- ========================================= -->
<!-- CLASS: Packet -->
<xsd:complexType name="Packet">
  <xsd:annotation>
    <xsd:documentation>Simple CCSDS Packet</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Header" type="CCSDS:Header">
      <xsd:annotation>
        <xsd:documentation>The packet header</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="DataField" type="CCSDS:DataField">
      <xsd:annotation>
        <xsd:documentation>The packet data field</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<!-- ========================================= -->
<!-- <<array>> Classes                         -->
<!-- ========================================= -->
<!-- CLASS: Header -->
<xsd:complexType name="Header">
  <xsd:annotation>
    <xsd:documentation>Sample packet header type</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="" maxOccurs="">
    <xsd:element name="item" type="Core:Octet"/>
  </xsd:sequence>
</xsd:complexType>
<!-- ========================================= -->
<!-- <<sequence>> Classes                      -->
<!-- ========================================= -->
<!-- CLASS: DataField -->
<xsd:complexType name="DataField">
  <xsd:annotation>
    <xsd:documentation>Sample packet data field type</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="item" type="Core:Octet"/>
  </xsd:sequence>
</xsd:complexType>
```

- CORBA IDL

```
#ifndef _EGOS2005_CCSDS_IDL_
#define _EGOS2005_CCSDS_IDL_

// This file is auto-generated by the EGOSMF CORBA IDL Generation Tool
// on Mon Nov 07 17:22:16 CET 2005
// from gen/idl/am-XASTRO-2_Framework_Example_EGOS_2005.xml
//
// THIS IS AUTO-GENERATED CODE - DO NOT EDIT

// ------------------------------------------------------------------------
// ----------------------- Types and Interfaces ---------------------------
// ------------------------------------------------------------------------

module CCSDS
{
    /// Sample packet data field type
    typedef sequence<octet, 65536> DataField;

    /// Sample packet header type
    typedef octet Header[6];

    /// Simple CCSDS Packet
    valuetype Packet
    {
        /// The packet header
        public Header header;

        /// The packet data field
        public DataField dataField;
    };
};

#endif /* _EGOS2005_CCSDS_IDL_ */
```

```
#ifndef _EGOS2005_PACKET_DECODER_IDL_
#define _EGOS2005_PACKET_DECODER_IDL_

// This file is auto-generated by the EGOSMF CORBA IDL Generation Tool
// on Mon Nov 07 17:22:16 CET 2005
// from gen/idl/am-XASTRO-2_Framework_Example_EGOS_2005.xml
//
// THIS IS AUTO-GENERATED CODE - DO NOT EDIT

// ------------------------------------------------------------------------
// ----------------------- Includes ---------------------------------------
// ------------------------------------------------------------------------

#include "../Core/PrimitiveTypes.idl"
#include "ccsds.idl"

// ------------------------------------------------------------------------
// ----------------------- Types and Interfaces ---------------------------
// ------------------------------------------------------------------------

/// Namespace for interface definitions
module CCSDS
{
    /// A packet decoder allows clients to submit packets to be decoded
    interface IPacketDecode
    {
        /// Error notification while decoding a packet
        exception DecodeFailed
        {
            Core::PrimitiveTypes::String8 reason;
        };

        /// Put the given packet into the decoder
        /// @param   packet the packet to decode
        /// @raises  DecodeFailed
        void Decode(
            in Packet packet
        ) raises (
            DecodeFailed
        );
    };
};
```

EGOS-MF code generation process

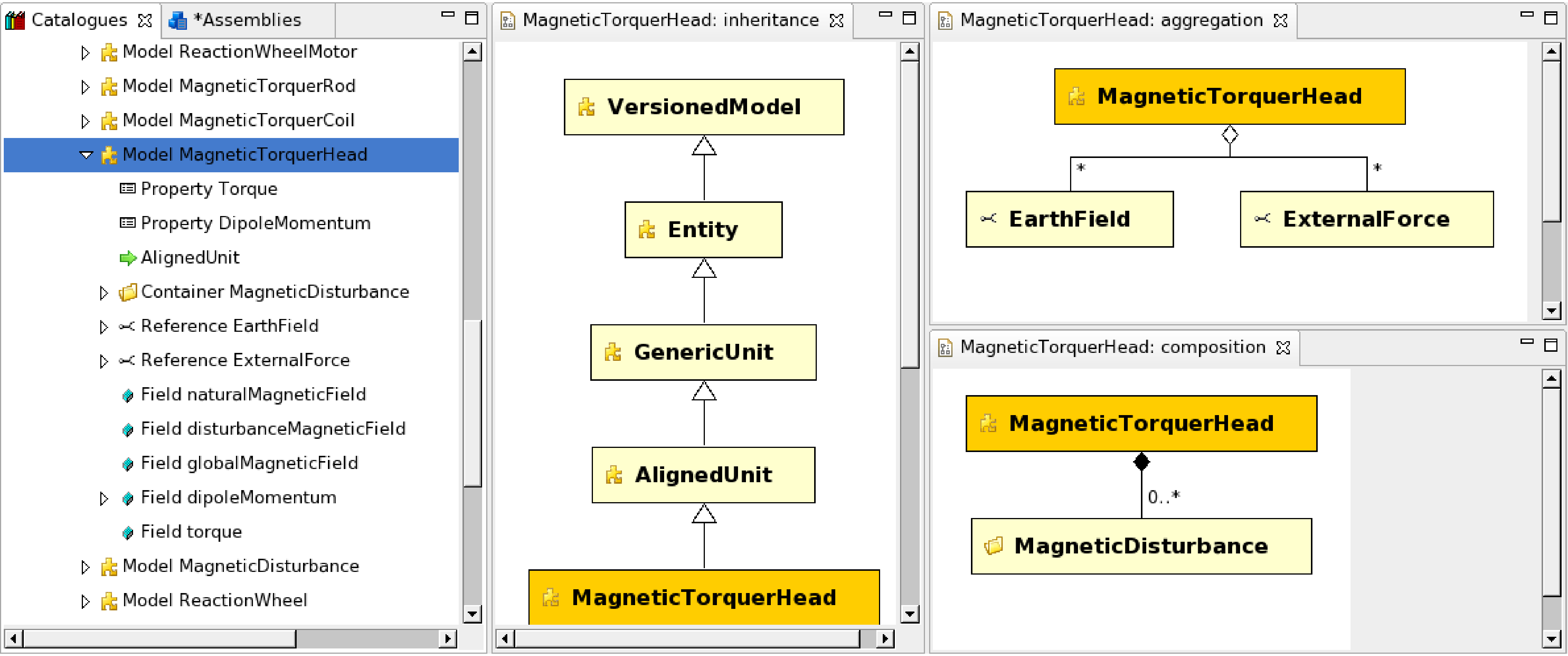


This systematic code generation process is illustrated next proceeding from the previous UML example

<<SMP2model>>MagneticTorquerHead

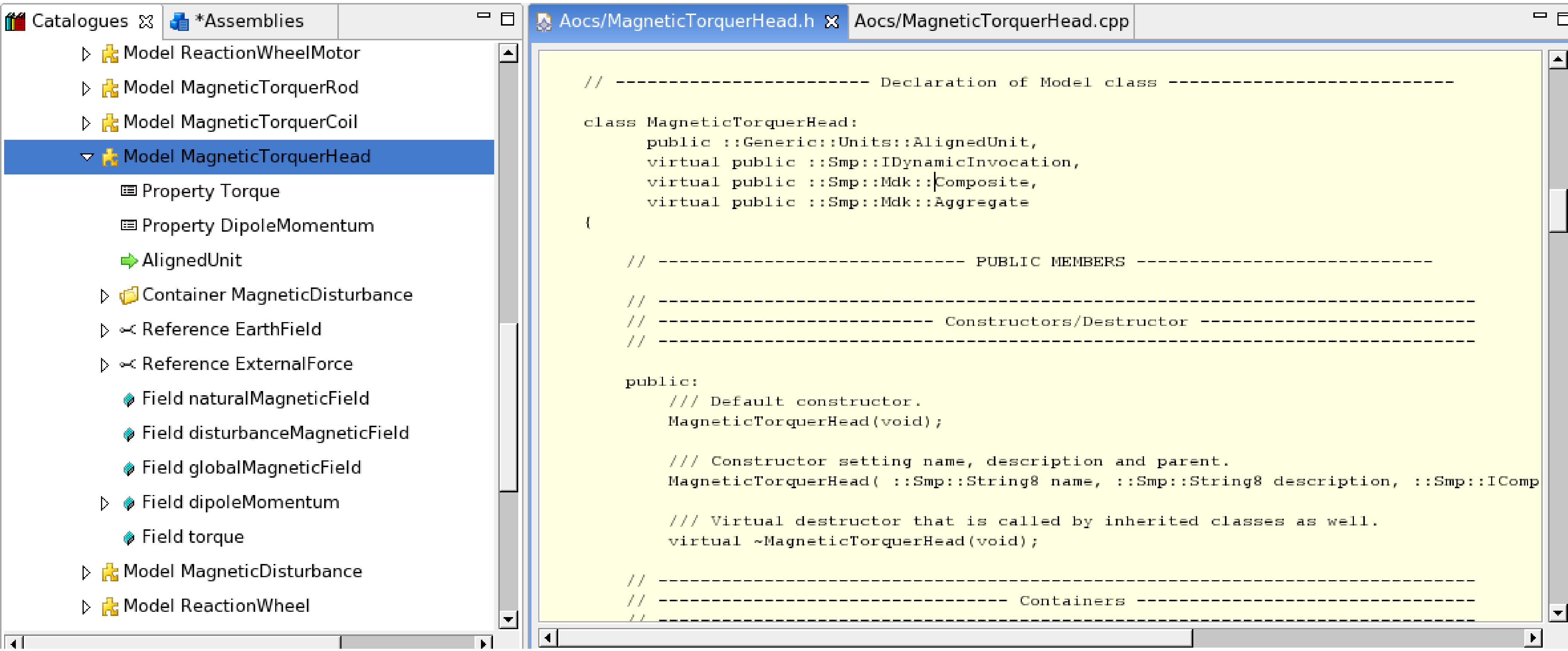
EGOS-MF example: Catalogue Generation

1. Catalogue Generator translated UML Design into SMP2 Catalogue (XML File)



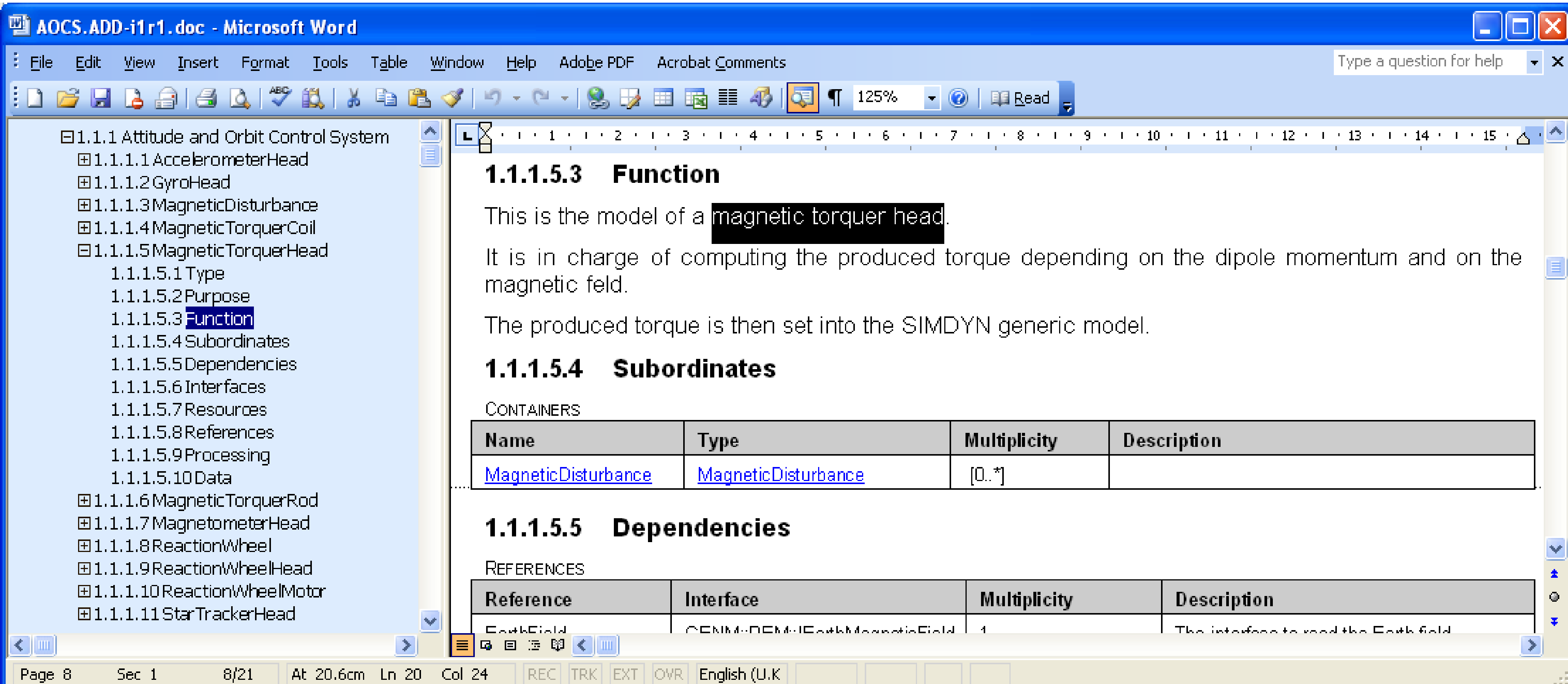
EGOS-MF example: C++ Code Generation

2. Code Generator generates C++ Source Code for each SMP2 Model wih SIMSAT MIE



EGOS-MF example: Document Generation

3. Document Generator generates Design Documentation for each SMP2 Model



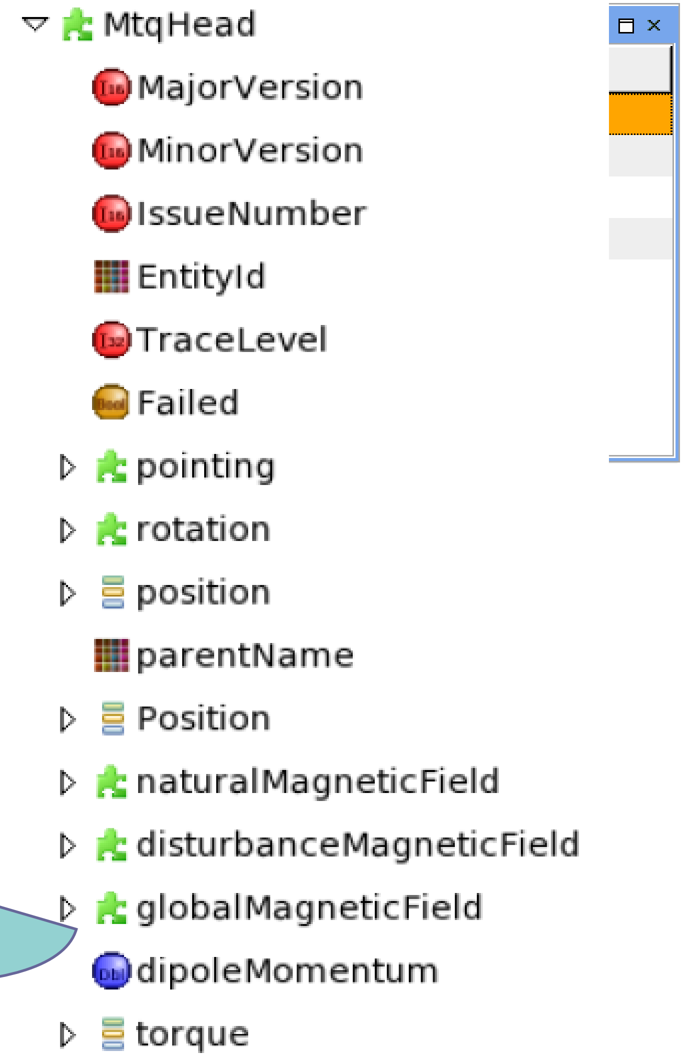
Users common views across simulators

1. Users have identical views on ...
   a. Message logging features
   b. Model failures
   c. Parameter limits
   d. Simulation tree search

Unknown AND Display [1]

#	Title	Type	Uni
0	GlobalMagneticField	Double	
1	softMinimum	Double	
2	softMaximum	Double	
3	hardMinimum	Double	
4	hardMaximum	Double	

2. Users have identical views (for example) on failed or f
   a. The magnetic torquer head has Analogue Param magnetic field) -- hence they have limits and th
   b. The magnetic torquer head can be failed becaus



Drag and drop directly from tree to display window

Conclusions

1. Building the ESA operational simulators is "better, faster, cheaper" with:
   a. SMP2 -- interfaces enabling reuse and exchange of satellite models
   b. REFA -- structured reuse of past acquired simulators knowledge
   c. EGOS-MF -- automatised component based development
   d. SIMSAT -- solid software infrastructure on linux platforms

2. From the users perspective
   a. These advanced technologies respond to project needs and schedules
   b. Requirements for multi-satellite systems are satisfied
   c. Demonstrated potential for reduced costs in producing future simulators

Thank you for your attention !

Any questions..??..

Max Pignède [*presenter*], Mario Merri, Vemund Reggestad

European Space Agency (ESA) / European Space Operations Centre (ESOC)

Peter Fritzen, Michael Irvine

VEGA Deutschland GmbH & Co. KG / A Finmeccanica Company