IEEE SOSE 2012
7th INTERNATIONAL CONFERENCE ON
SYSTEM OF SYSTEMS ENGINEERING
July 16-19, 2012 – Genoa, Italy

Modeling and Simulation for System Reliability Analysis: The RAMSAS Method

Alfredo Garro — *PhD, Associate Professor*

Andrea Tundis — *PhD Student*

Systems Engineering and Integration *(SEI)* ***Research Group***

Department of Electronics, Computer and System Sciences (D.E.I.S.)
University of Calabria – ITALY

Outline

❶ System Reliability Analysis

❷ A Model-Based method for System Reliability Analysis

❸ From Large-scale Systems to System of Systems

❹ Conclusions and future works

System Dependability and RAMS Analysis

- **Dependability**: **“**the collective term used to describe the **availability** performance and its influencing factors: **reliability** performance, **maintainability** performance and maintenance support performance**”** (*IEC - International Electrotechnical Commission*)

- **RAMS** (*Reliability*, *Availability*, *Maintainability* and *Safety*): the engineering discipline which aims at providing an integrated and methodological approach to deal with system dependability

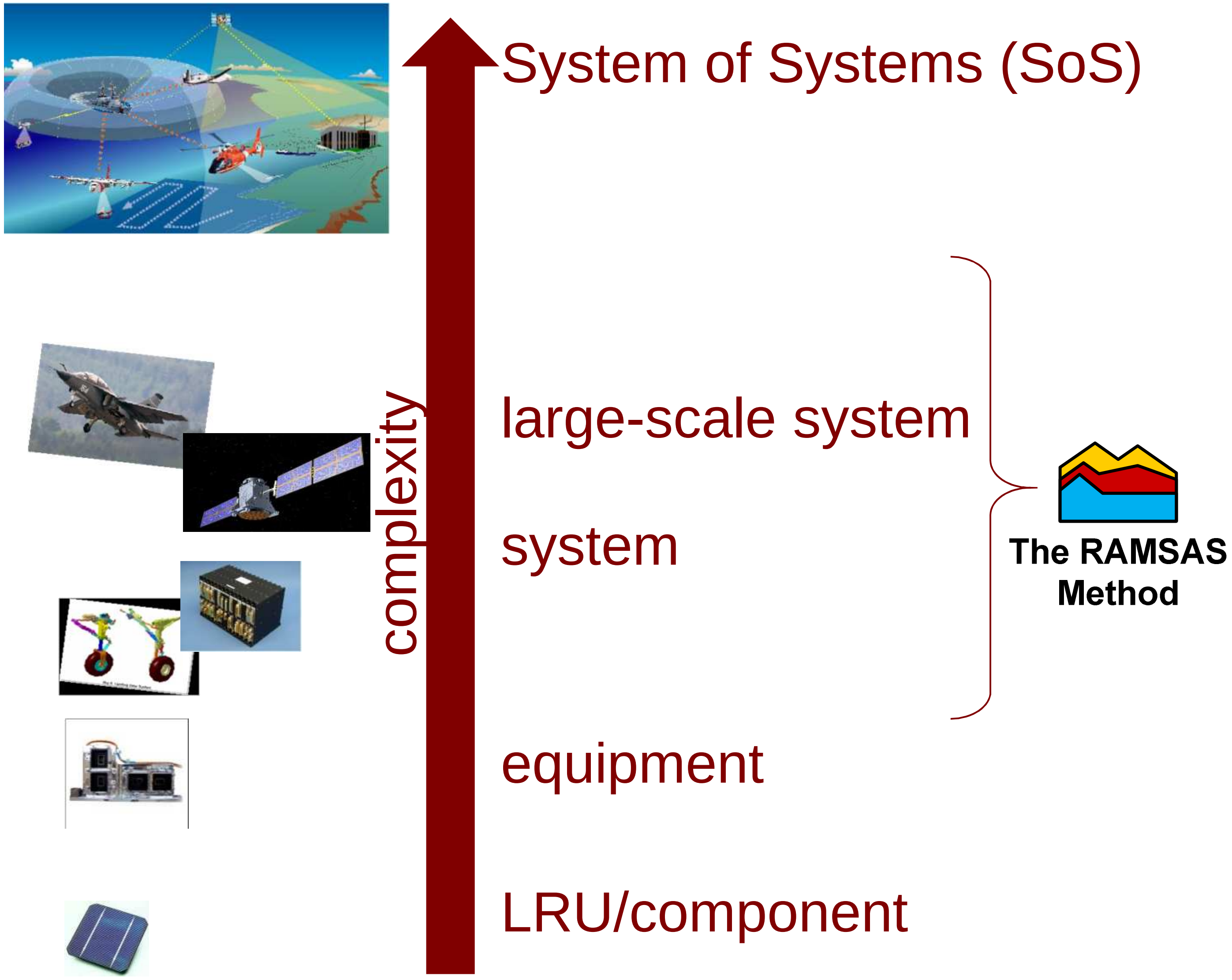
System Dependability and RAMS Analysis

- The main objective of RAMS analyses is to identify causes and consequences of system **failures**
- RAMS analyses are typically carried out using a **layered approach** and through both ***quantitative*** and ***qualitative*** **analysis techniques** as:
  - *series-parallel* system reliability analysis
  - *Markov Chain models*
  - *FMECA* (*Failure Modes Effects and Critical Analysis*)
  - *FTA* (*Fault Tree Analysis*)
  - ….

System Dependability and RAMS Analysis

	Quantitative Analysis	**Qualitative Analysis**	**Suitable for Software Intensive Systems**
Series-Parallel (RBD)	x	-	-
Markov Chains	x	-	-
FMEA/FMECA	-	x	x *(S-FMEA/S-FMECA)*
FTA	-	x	x *(S-FTA)*
HAZOP	-	x	x
HSIA	-	x	x
SCCFA	-	x	x
PSH	-	x	x

Reliability Analysis: from LRUs (Lowest Replaceable Unit) to SoS (System of Systems)



System of Systems (SoS)

large-scale system

system

equipment

LRU/component

complexity

The RAMSAS Method

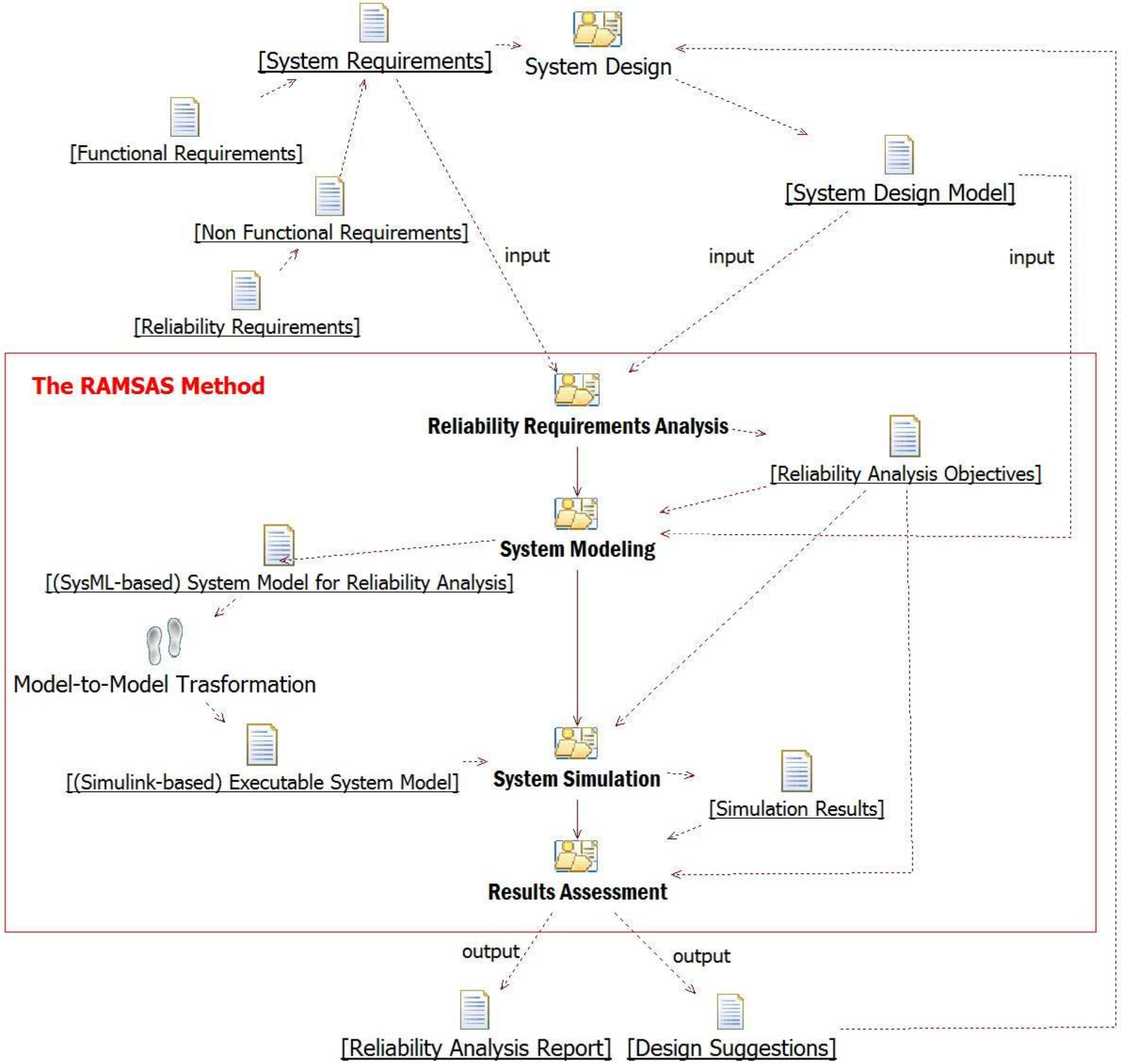
RAMSAS: A Model-Based method for System Reliability Analysis

- centered on a popular UML-based language for system modeling (***SysML***)
- exploiting a *de facto* standard platform for the simulation of multi-domain dynamic and embedded systems (*Mathworks* ***Simulink***)
- fully specified as a ***method*** (in terms of *phases*, input and output *workproducts*, etc.) and thus “pluggable” in a complete System Development Process (e.g. based on a V-Model)

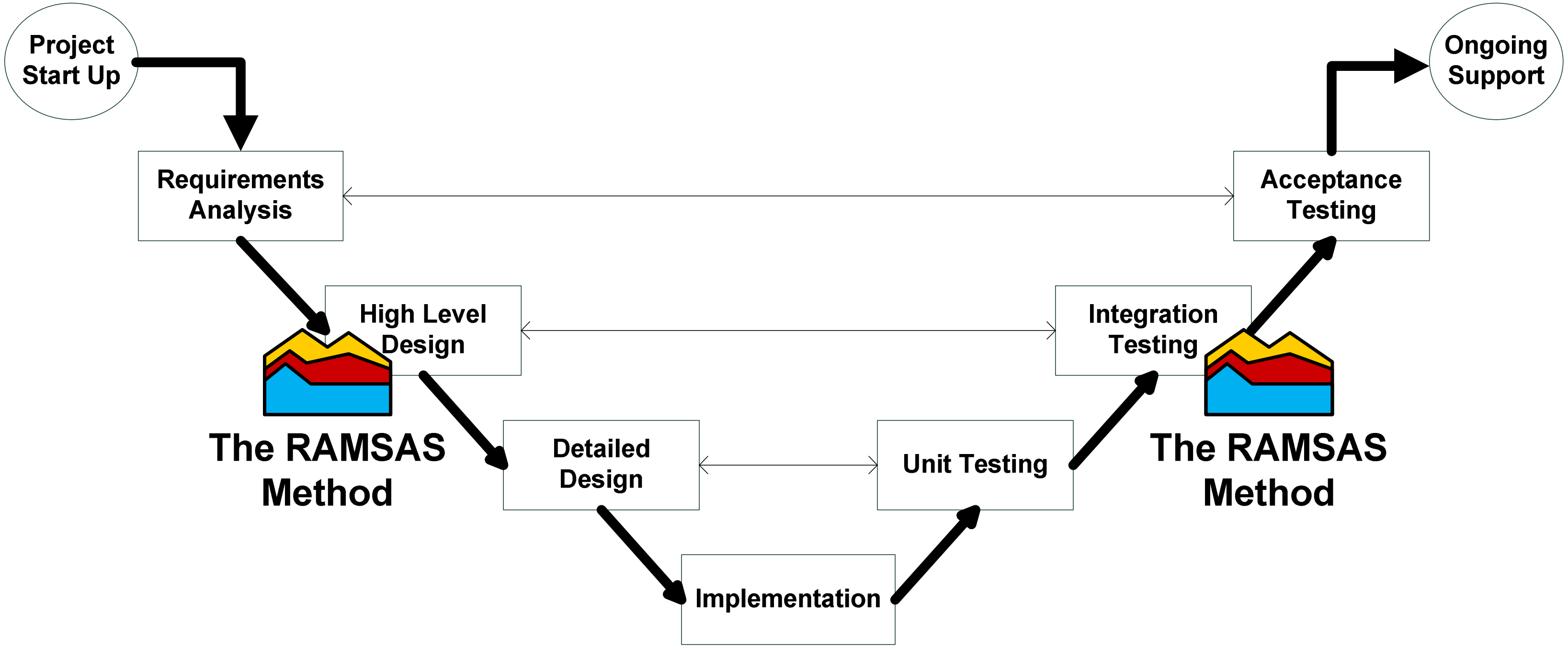
RAMSAS: A Model-Based method for System Reliability Analysis

The **RAMSAS** method is centered on a classical **iterative process** which consists of four main phases:

- ***Reliability Requirements Analysis***
- ***System Modeling***
- ***System Simulation***
- ***Results Assessment***


[System Requirements]
System Design
[Functional Requirements]
[Non Functional Requirements]
[Reliability Requirements]
[System Design Model]
input
input
input
The RAMSAS Method
Reliability Requirements Analysis
[Reliability Analysis Objectives]
System Modeling
[(SysML-based) System Model for Reliability Analysis]
Model-to-Model Trasformation
[(Simulink-based) Executable System Model]
System Simulation
[Simulation Results]
Results Assessment
output
output
[Reliability Analysis Report]
[Design Suggestions]

When and where to exploit our method in a typical System Development Process



The proposed method is not intended to be an *alternative* to other RAMS techniques (FMECA, FTA, RDB, etc.) but rather a **complement** able to provide

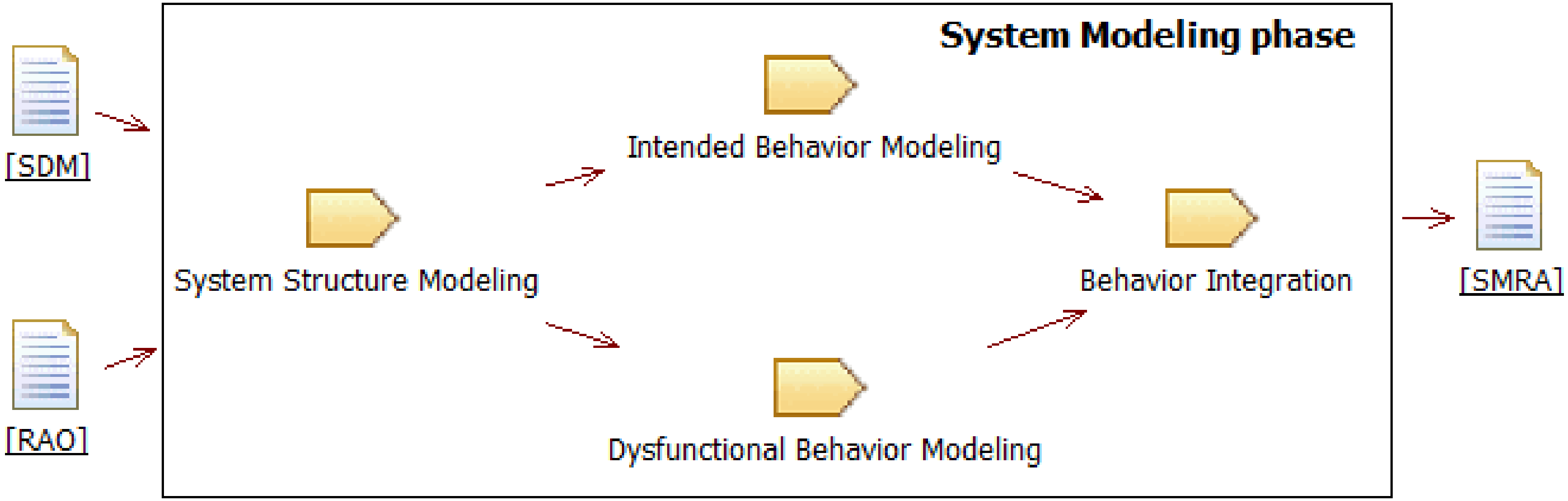
additional analysis capabilities

RAMSAS: The ***Reliability Requirements Analysis*** phase

- In the ***Reliability Requirements Analysis*** phase, the objectives of the system reliability analysis are specified.
- INPUT work-products: System Design, System Requirements (functional and non-functional) + a Failure Modes and Effects Analysis (FMEA)
- OUTPUT work-products: *Reliability Analysis Objectives*
  - The **functions** that the system has to perform, the related **operative conditions**, and the reference **time horizons** must be clearly individuate along with the main **system failures** and their **local** and **global effects**
  - The **reliability** functions and **indicators**, to be derived from the analysis of the simulation results, must be identified along with the main **analysis techniques** to be applied to the data gathered from simulation

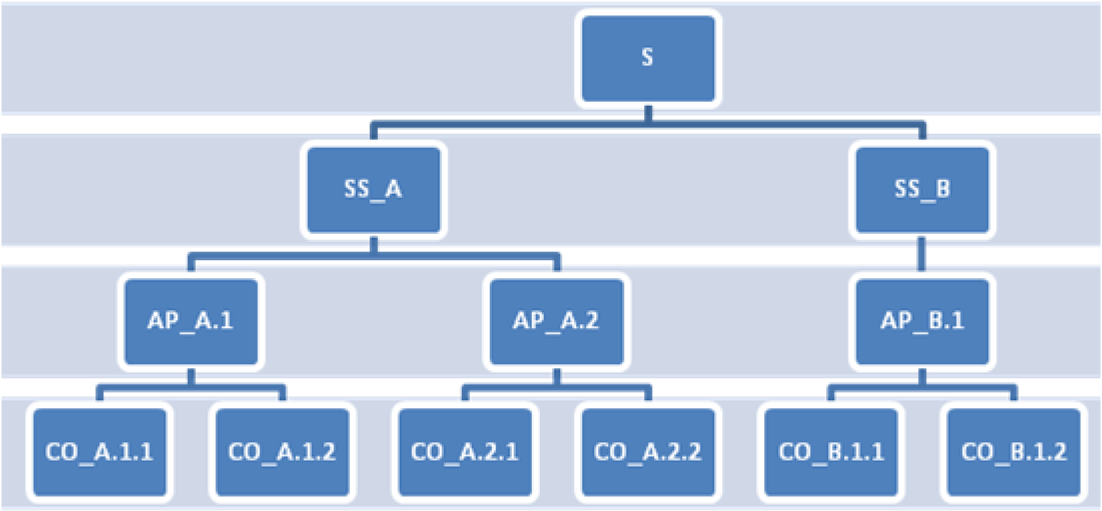
RAMSAS: The ***System Modeling*** phase

- In the ***System Modeling*** phase the **structure** and both the **intended** and **dysfunctional behavior** of the System under consideration are modeled by using a ***SysML*** based notation.


System Modeling phase
[SDM]
[RAO]
System Structure Modeling
Intended Behavior Modeling
Dysfunctional Behavior Modeling
Behavior Integration
[SMRA]

RAMSAS: The ***System Modeling*** phase → *System Structure Modeling*

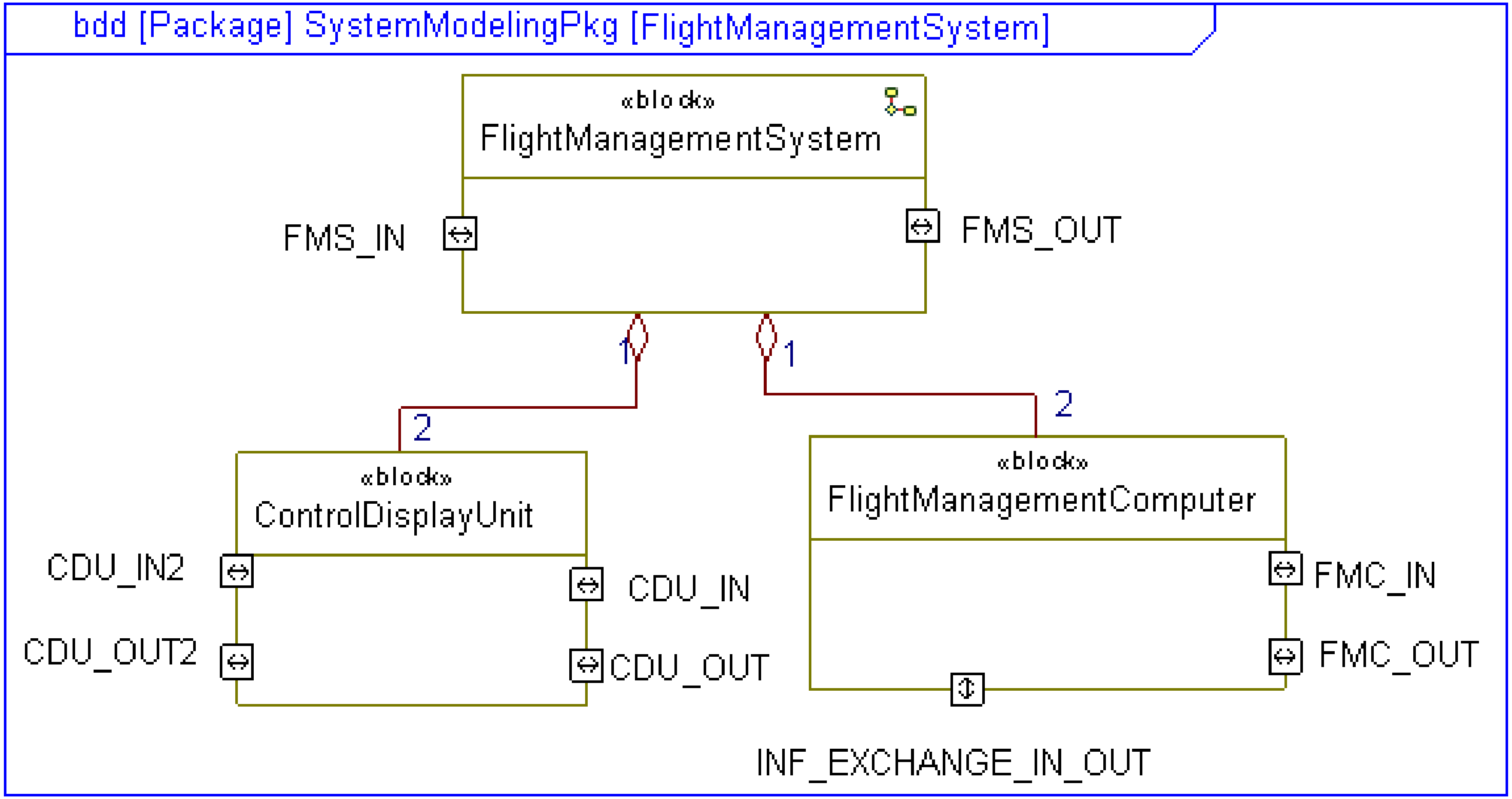
- In this phase the System is **decomposed** in component entities by applying ***in-out zooming mechanisms***.



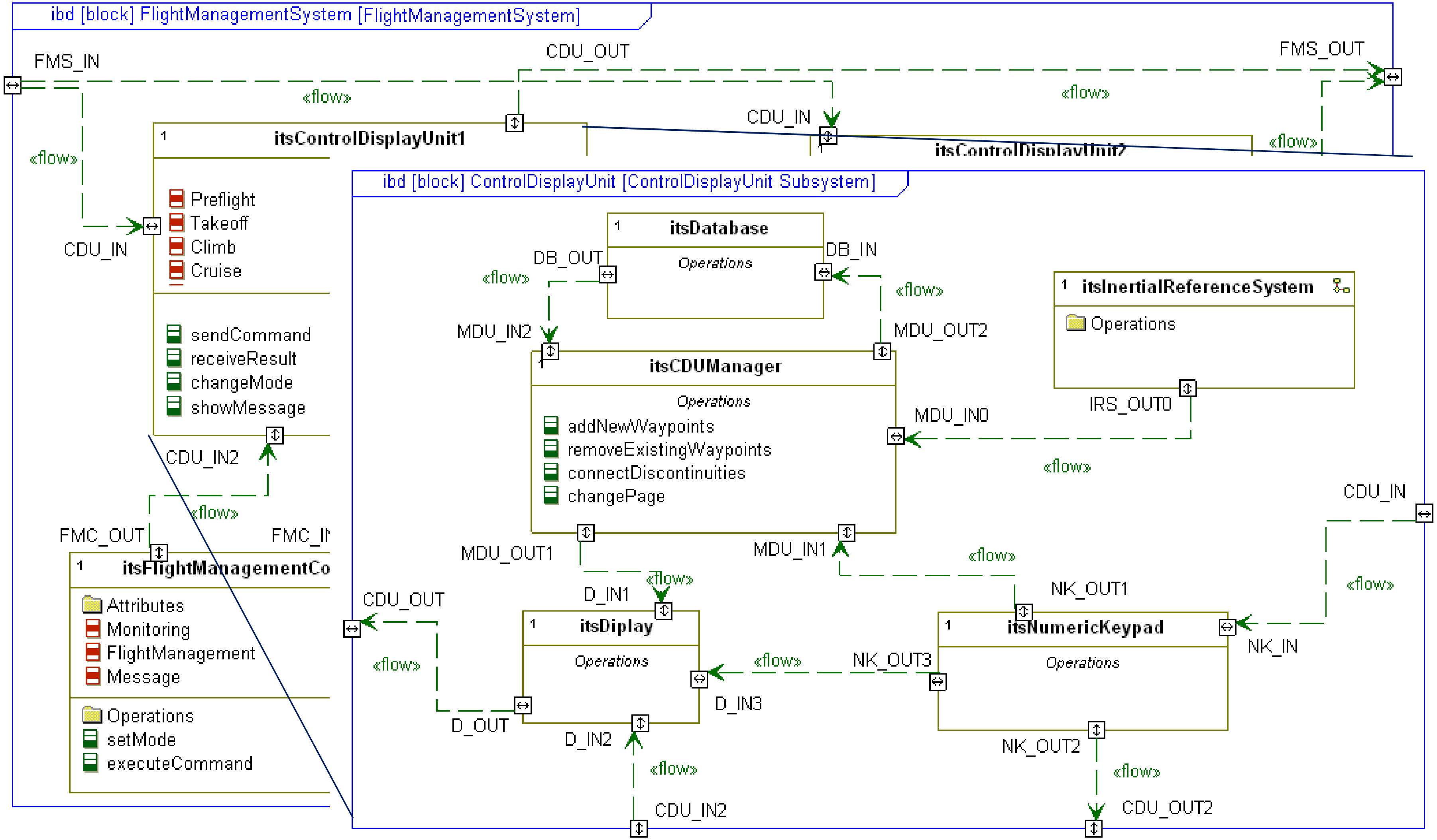
RAMSAS: The ***System Modeling*** phase → *System Structure Modeling*

- **Each system entity is modeled as a SysML *Block***
- for a given abstraction level:
  - a *Block Definition Diagram* (**BDD**) describes a block with its *interfaces*, *attributes*, *operations*, *constraints*, *parts* and *relationships* with other blocks;
  - an *Internal Block Diagram* (**IBD**) provides a description of the block internal structure in terms of the *organization* of its component blocks

RAMSAS: The ***System Modeling*** phase → *System Structure Modeling*



RAMSAS: The ***System Modeling*** phase → *System Structure Modeling*



RAMSAS: The ***System Modeling*** phase → *Intended Behavior Modeling*

- The reference model is **service** and **task-oriented**:
  - the behavior of each entity is modeled in terms of the services (or functions) that the entity is able to provide and which are performed through tasks.
- In order to specify the behavior of the system and its component entities, two levels of decomposition are considered:
  - ***leaf level*** (e.g. component level)
  - ***non-leaf level*** (e.g. equipment, subsystem or system level)

RAMSAS: The ***System Modeling*** phase → *Intended Behavior Modeling*

- for **each entity at the leaf decomposition level** (the lowest decomposition level):
  - the **services** (or functions) provided by the entity should be specified;
  - **each task** (flow of activities/actions) performed by the entity for providing a specific service (or function) has to be specified through an **Activity Diagram**;
  - the **exchange of messages** between the entity and the external environment should be represented through **Sequence Diagrams**;
  - in case the behavior of the entity depends on its internal state, a state machine which models the **entity life cycle** can be specified through a **Statechart Diagram**.

RAMSAS: The ***System Modeling*** phase → *Intended Behavior Modeling*

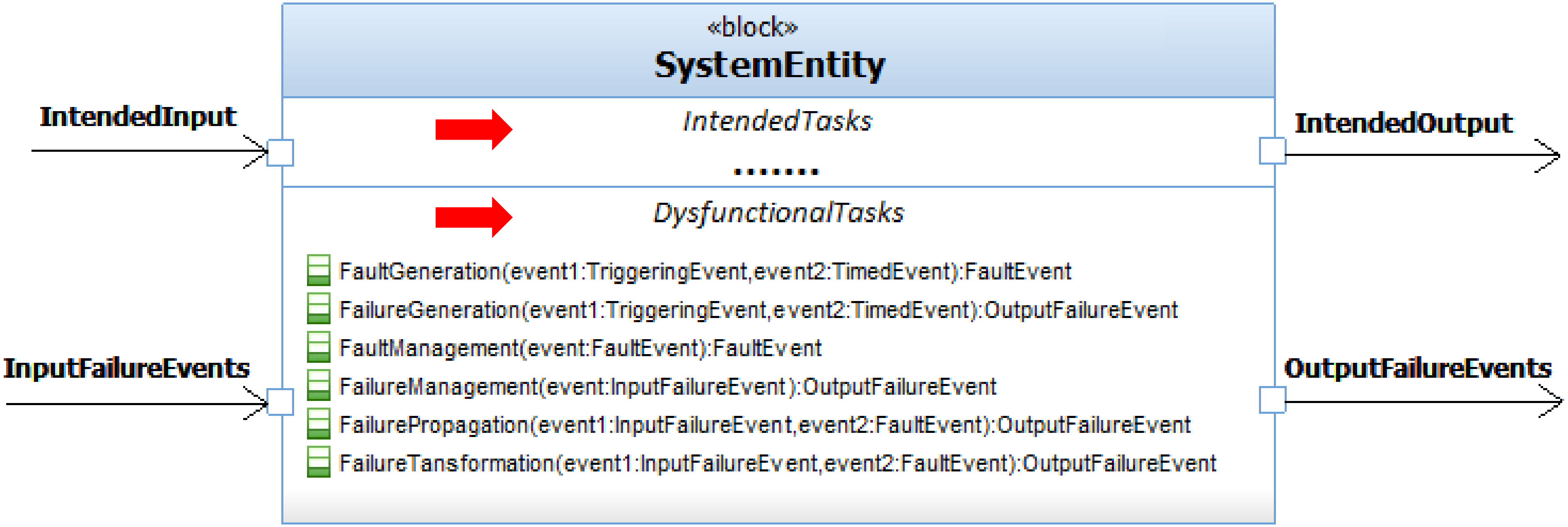
- at higher decomposition levels (**non-leaf decomposition levels**), the representation of the entity behavior is similar… but….
  - in modeling the entity tasks through **Active Diagrams** the **responsibilities of each sub-entity** should be highlighted through **swimlanes**;
  - In modeling the exchange of messages between the entity and the external environment through **Sequence Diagrams** the **sub-entities** should be introduced as **participants** in the diagrams;
  - the **Statechart Diagram** which models the life-cycle of the entity can adopt **advanced constructs** (AND/OR decomposition etc.) for representing **how the behavior of the entity is related to the behavior of its sub-entities**

RAMSAS: The ***System Modeling*** phase → *Intended Behavior Modeling*

The modeling of the intended behavior can be straightforward if during the system design similar structural and behavioral reference models have been adopted along with a UML based modeling notation

RAMSAS: The ***System Modeling*** phase → *Dysfunctional Behavior Modeling*

- In the Dysfunctional Behavior Modeling activity, the focus is on the modeling of **faults** (*a defect in a block*) and **failures** (*an observable deviation from the intended behavior at the system boundary*)



«block»
SystemEntity

IntendedTasks

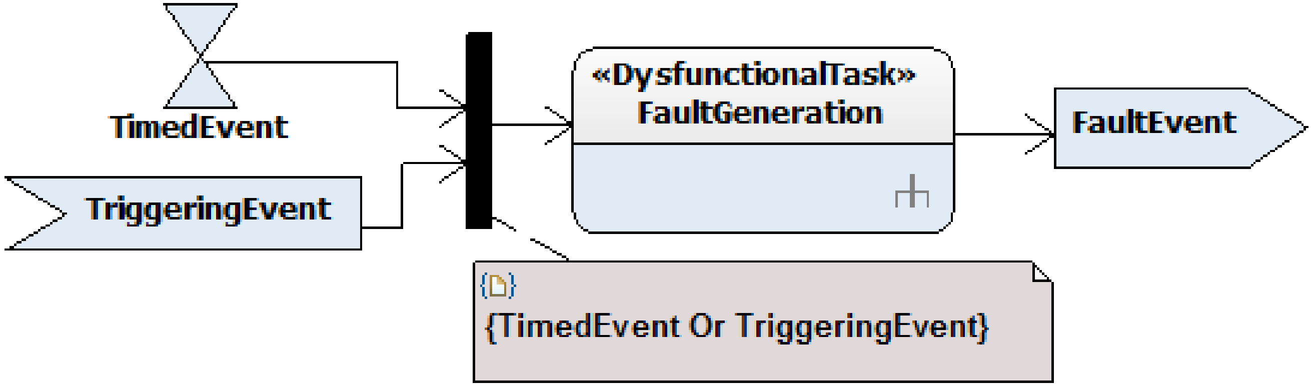
....

DysfunctionalTasks

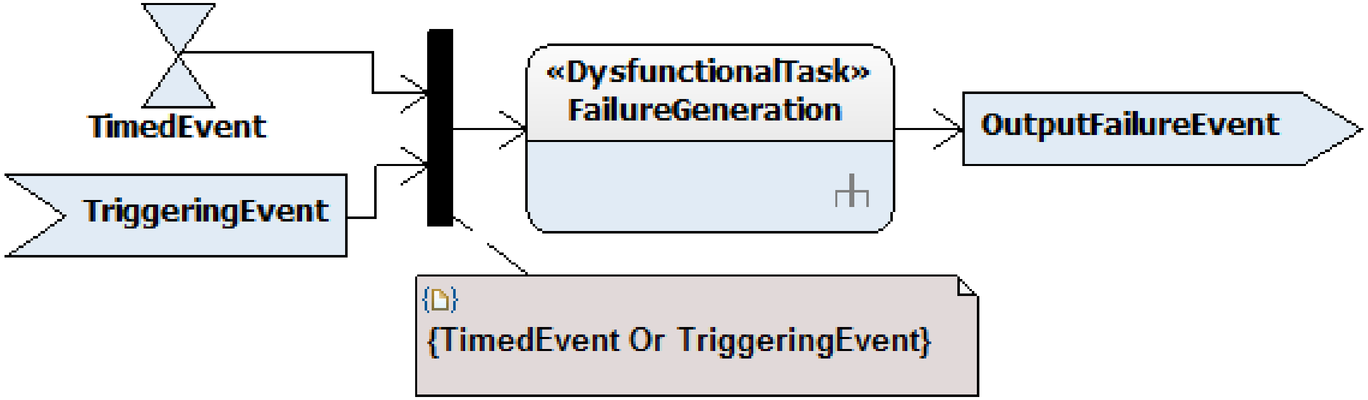
- FaultGeneration(event1:TriggeringEvent,event2:TimedEvent):FaultEvent
- FailureGeneration(event1:TriggeringEvent,event2:TimedEvent):OutputFailureEvent
- FaultManagement(event:FaultEvent):FaultEvent
- FailureManagement(event:InputFailureEvent):OutputFailureEvent
- FailurePropagation(event1:InputFailureEvent,event2:FaultEvent):OutputFailureEvent
- FailureTansformation(event1:InputFailureEvent,event2:FaultEvent):OutputFailureEvent

RAMSAS: The ***System Modeling*** phase → *Dysfunctional Behavior Modeling*

- Six templates of *dysfunctional tasks* have been individuated:



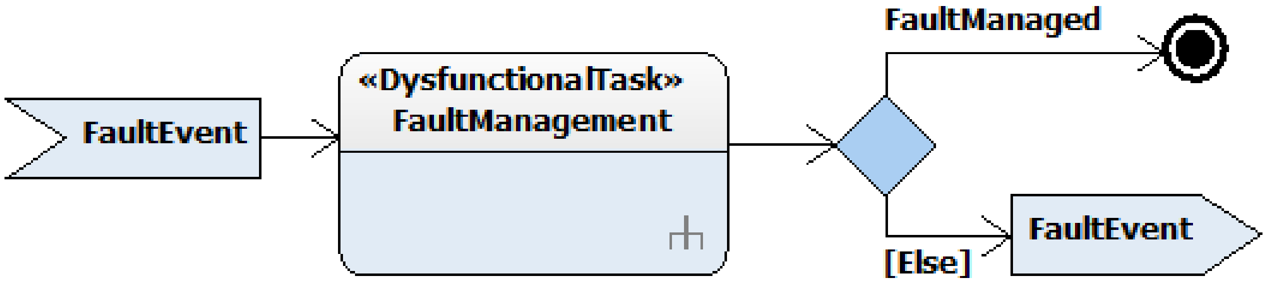
a) Fault Generation



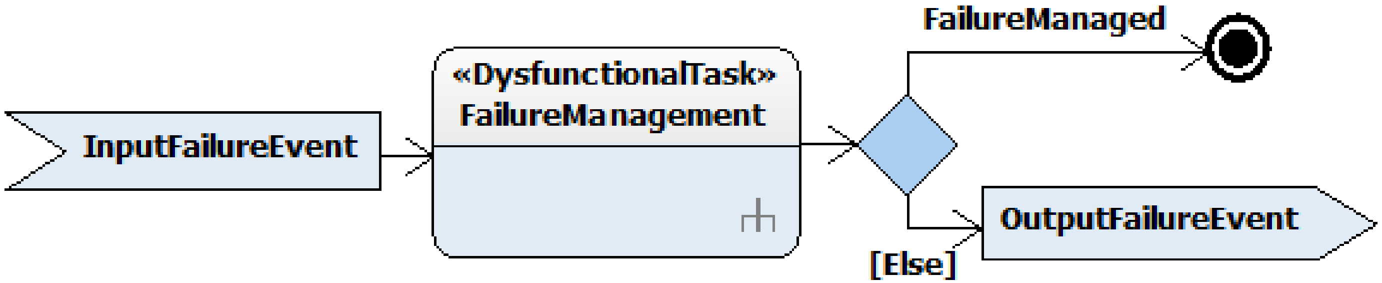
b) Failure Generation

RAMSAS: The ***System Modeling*** phase → *Dysfunctional Behavior Modeling*

- Six templates of *dysfunctional tasks* have been individuated:



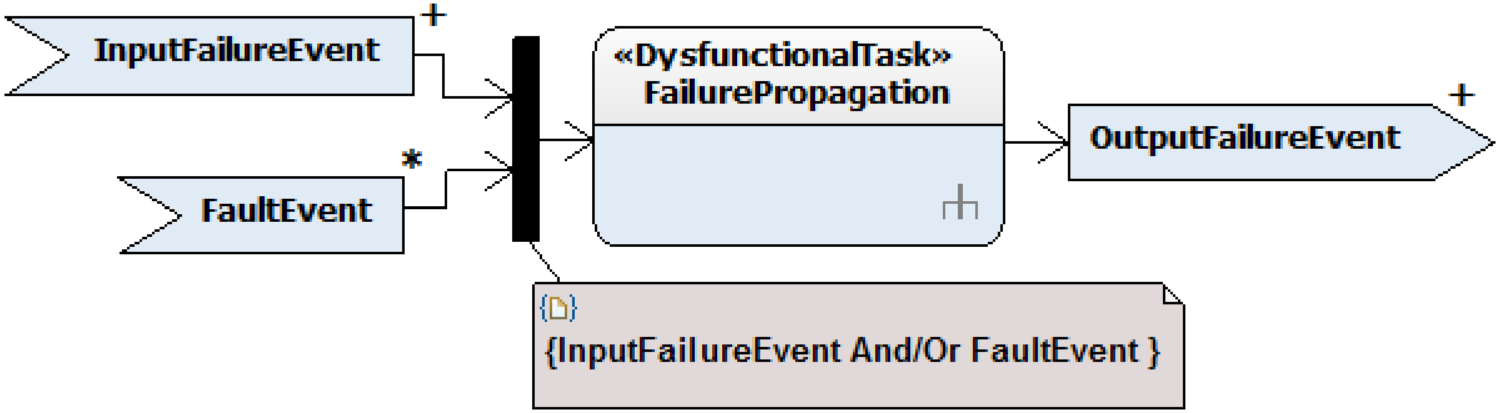
d) Fault Management



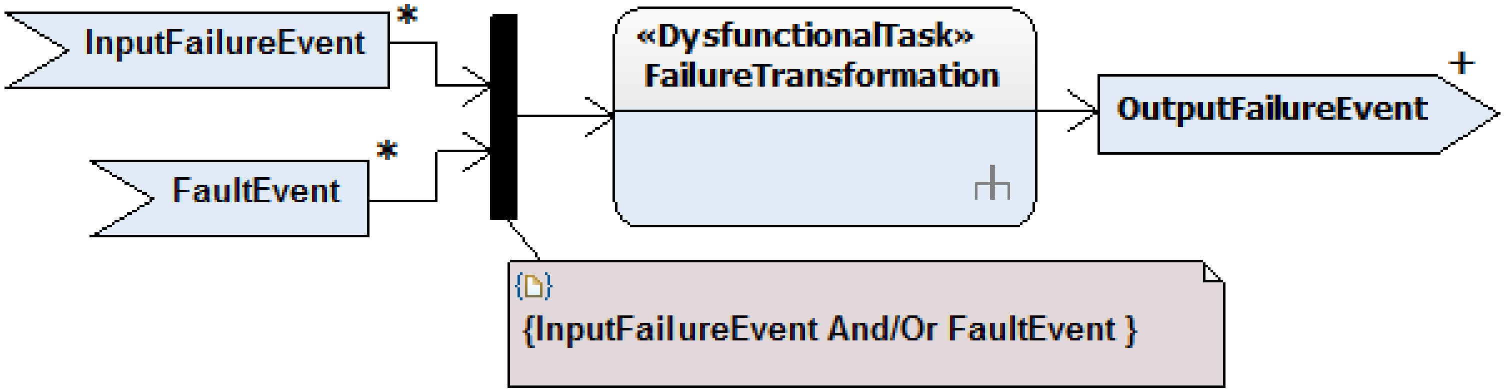
c) Failure Management

RAMSAS: The ***System Modeling*** phase → *Dysfunctional Behavior Modeling*

- Six templates of *dysfunctional tasks* have been individuated:



e) Failure Propagation



f) Failure Transformation

RAMSAS: The ***System Modeling*** phase → *Dysfunctional Behavior Modeling*

To further support this crucial modeling activity, a set of ***patterns*** to associate to each of the above discussed six types of *dysfunctional tasks* should be defined.

A basic pattern is associated to a couple

(dysfunctional task type; fault/failure type)

As a consequence, beside the individuated six dysfunctional task types, **a** (possibly hierarchical) **classification of faults/failures needs to be introduced**.

RAMSAS: The ***System Modeling*** phase → *Dysfunctional Behavior Modeling*

A first solution could consider the following fault/failure types:

(i) *reaction too late*;

(ii) *reaction too early*;

(iii) *value failure*;

(iv) *commission*;

(v) *omission*.

By **combining** the individuated **six dysfunctional task types** with these **five fault/failure types**, **thirty different basic fault/failure behavioral patterns** **can be defined**

RAMSAS: The ***System Modeling*** phase → *Behavior Integration*

intended behaviors + dysfunctional behaviors

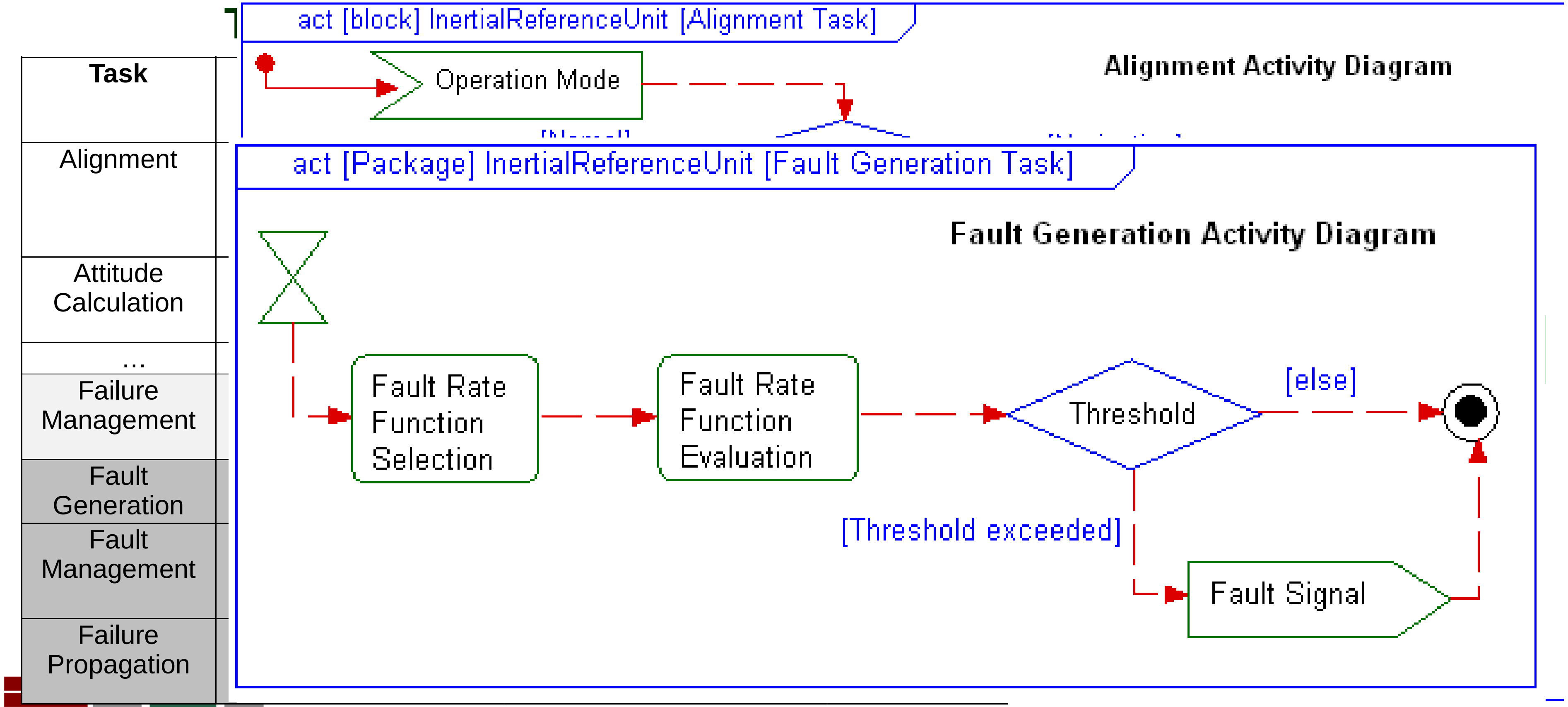
⬇

an overall behavioral model of the system and its component entities

This activity closes the System Modeling phase by delivering the *System Model for Reliability Analysis* (*SMRA*) work-product

RAMSAS: The ***System Modeling*** phase → *Behavior Integration*

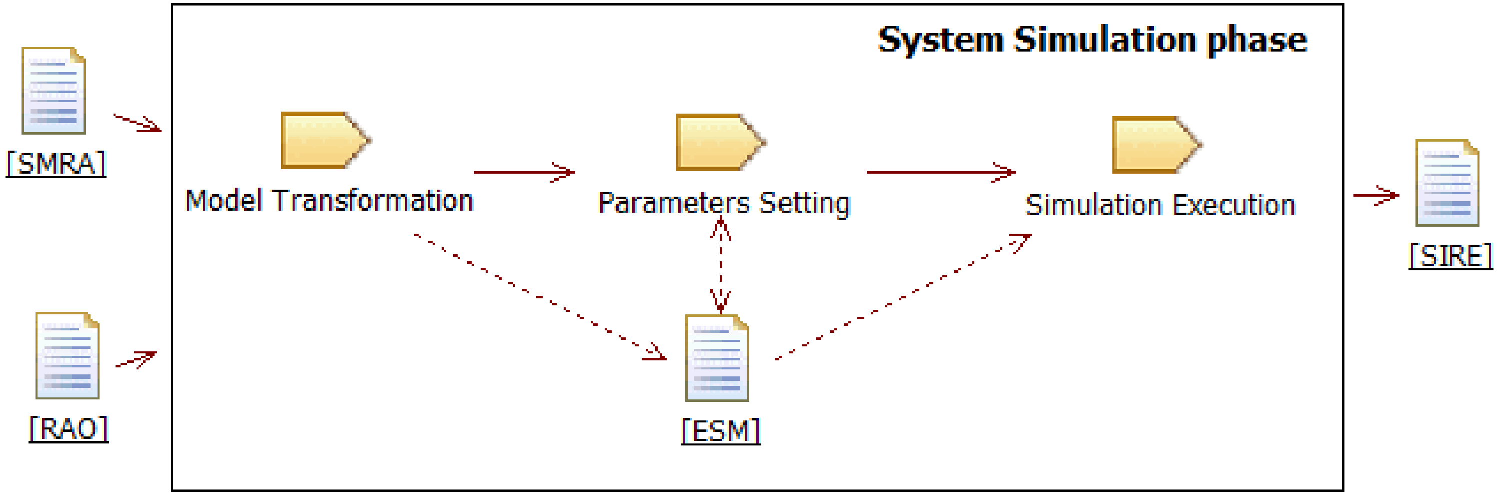
- An example of behavioral modeling: the specification of the behavior of an *Inertial Reference Unit* , a key component of the IRS of a Control Display Unit (CDU) of a FMS:



Task
Alignment
Attitude Calculation
…
Failure Management
Fault Generation
Fault Management
Failure Propagation
act [block] InertialReferenceUnit [Alignment Task]
Alignment Activity Diagram
Operation Mode
act [Package] InertialReferenceUnit [Fault Generation Task]
Fault Generation Activity Diagram
Fault Rate Function Selection
Fault Rate Function Evaluation
Threshold
[else]
[Threshold exceeded]
Fault Signal

RAMSAS: The ***System Simulation*** phase

- The objective of the ***System Simulation*** phase is to evaluate through simulation the reliability performance of the system and, possibly, compare different design alternatives and parameters settings



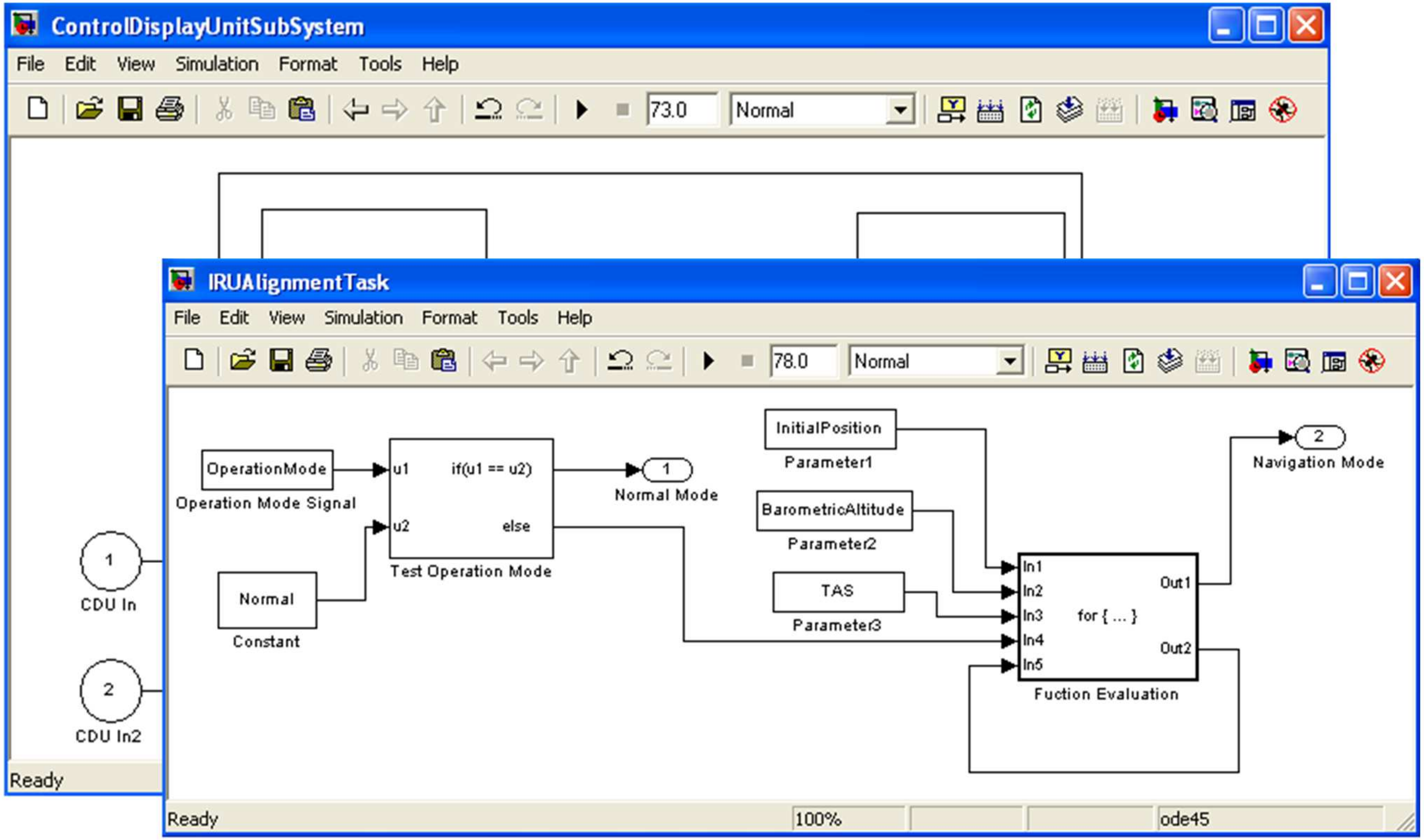
RAMSAS: The ***System Simulation*** phase

- **Transformation between models is based on a mapping between the basic SysML and Simulink constructs**:

Entity	SysML	Simulink
System/Subsystem/ Equipment/Component	Block, Part	Block, Subsystem Block
Behavior/Constraint	Activity Diagram, Sequence Diagram, Statechart Diagram, Parametric Diagram	S-Function, State Flow diagram
Input/Output Interface	Flow Port	Input/Output Simulink Block
Association/Binding	Connection	Line

The *Mealy Machines* which model the behavior of a **Simulink Block** is obtained by the corresponding **SysML Behavioral Diagrams**

RAMSAS: The ***System Simulation*** phase



RAMSAS: The ***Results Assessment*** phase

- In the ***Results Assessment*** phase, the **data gathered from the simulations are analyzed with reference to the objectives of the reliability analysis identified in the initial phase of the process:**
  - directly in **Simulink**
  - by using useful add-on like ***SIMLOG***
  - by **external analysis tools**

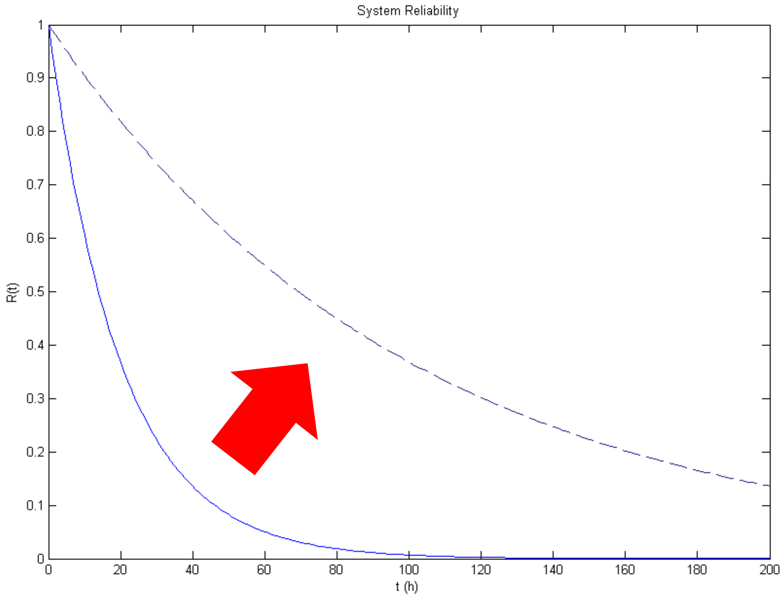
RAMSAS: The ***Results Assessment*** phase

As a result, the following two work-products are produced in output:

- *Reliability Analysis Report* (*RAR*), which provides a detailed analysis about the reliability performance of the system under consideration;
- *Design Suggestions Report* (*DSR*), which provides a set of suggestion to improve the design of the system and/or choose among different design choices.

- As for any iterative process, **new (partial or complete) iterations can be executed for achieving new or missed analysis objectives**

RAMSAS: The ***Results Assessment*** phase



- The analysis of the simulation data can provide **useful indications** which allow obtaining a **more descriptive and predictive reliability system model** and **suggest some design choices which could improve the system reliability indicators**

Exploiting the RAMSAS method for System Reliability Analysis

- **RAMSAS** has been already experimented
  - in the **avionics** domain for the reliability analysis of:
    - a **Landing Gear System** [1];
    - a **Flight Management System** [2];
  - in the **automotive** domain for the reliability analysis of an **Anti-lock Brake System** (ABS) [3];
- an ongoing experimentation concerns the reliability analysis of an **Attitude Determination and Control System** (ADCS) of a **satellite** [4]
- [1] A. Garro, A. Tundis, and N. Chirillo, “System reliability analysis: a Model-Based approach and a case study in the avionics industry,” Proc. of the 3rd Air and Space International Conference (CEAS), Venice (Italy), October 2011.
- [2] A. Garro and A. Tundis, “A Model-Based method for System Reliability Analysis,” Proc. of the Symposium On Theory of Modeling and Simulation (TMS), Orlando, FL (USA), March 2012.
- [3] A. Garro and A. Tundis, “Enhancing the RAMSAS method for System Reliability Analysis: an exploitation in the automotive domain,” Proc. of the 2nd Int. Conf. on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), Rome (Italy), July 2012.
- [4] A. Garro, A. Tundis, J. Groß , and M. Riestenpatt Gen. Richter, " Experimenting the RAMSAS method in the reliability analysis of an Attitude Determination and Control System (ADCS)," Proc. of the Int. Workshop on Applied Modeling and Simulation (WAMS)", Rome (Italy), September, 2012.

From Large-scale Systems to System of Systems

- The current version of RAMSAS has been originally conceived for **large-scale systems** (e.g. military and commercial aircraft, spacecraft, satellites, power plant automobiles,…)
- The reliability analysis of these systems is a challenging task which, as proved by RAMSAS, could benefit both from **model-based approaches** and from **simulation**
- However, although the **structure** of these large-scale systems is rather complex (or better complicated) it **remains** quite **the same during the system life cycle**; moreover, a great part of the system components manifest a reactive behavior and **pro-activeness is limited to a narrow subset of components**

From Large-scale Systems to System of Systems

- The **reliability analysis of a SoS presents different and peculiar aspects respect to that of a large-scale systems**...
- Some features of RAMSAS are particularly suited for the reliability analysis of SoS:
  - the adoption of **zooming in-out mechanisms** for the **structural** and **behavioral modeling** of the system;
  - the exploitation of **simulation** both for the **analysis** of system **properties** and for the **evaluation** of **alternative scenarios** and **design choices**.

From Large-scale Systems to System of Systems

- However, new features need to be added:
  - the possibility to define the **potential changes in the system structure** during the system life cycle
  - specific concepts to explicitly **model the pro-active** (and thus autonomous) **part of the behavior of the SoS entities** (which are themselves systems), such as the goals that the entity will to achieve….
- These novelties could lead to **evolve the current vision of system adopted in RAMSAS to a more agent-oriented one** in which a **SoS is treated, and thus modeled and simulated as a Multi-Agent System**…

Conclusions and Future Work

- RAMSAS combines in a unified framework the strengths of **powerful visual modeling languages** (as OMG SysML) with **mature and popular simulation tools** (as Mathworks Simulink)
- The proposed method is not intended to be an ***alternative*** to other RAMS techniques (FMECA, FTA, RDB, etc.) but rather a **complement** able to provide additional analysis capabilities
- The **method** can be integrated in various phases of a typical System Development Process (e.g. in the **Verification** and/or **Design phases** of a V-Cycle)
- This allows supporting the **satisfaction** and **traceability of an important non-functional requirement, such as reliability, in the early stages of a development process** with considerable **time and cost reductions** respect to more traditional reliability analyses techniques which are often carried on in the last stages of the development with the risk of having to revise even basic design choices.

Conclusions and Future Work

Ongoing research efforts are devoted to:

- **enrich and improve** RAMSAS
- **extensively experiment** RAMSAS in the analysis of mission-critical systems in different application domains
- **integrate** RAMSAS in the IBM *Rational Harmony for Systems Engineering* process
- **support** other environments for carrying out the Simulation Phase (e.g. OpenModelica)
- **investigate** how to further extend RAMSAS so as to effectively support the reliability analysis not only of large-scale systems but also of SoS…

Acknowledgments

- *Andrea Tundis* (SEI Research Group, University of Calabria)
- *Johannes Groß, Marius Riestenpatt gen. Richter* (Institute for Statics and Dynamics of Aerospace Structures University of Stuttgart)
- *Peter Fritzson , Olena Rogovchenko* (PELAB, Linköping University)
- *Henry Broodney*, *Michael Masin* (IBM Haifa Research Center)
- *Daniele Gianni* (ESA-ESTEC)
- *Gabriele Luceri*, *Nicola Chirillo* (Z-Lab Engineering)

Thank you

garro@deis.unical.it