



# Business Case for Model-Based Systems Engineering (MBSE) with SysML

*Metrics from an Aerospace Industry Case Study*

Presenter:

Lenny Delligatti, M.S., CSEP, OCSMP, OCUP



Last modified: 27 JAN 2014



# Overview of MBSE

“MBSE is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.”

-- International Council on Systems Engineering (INCOSE),  
*Systems Engineering Vision 2020*, Sep 2007



# Overview of MBSE:

## *The System Model*

The output of MBSE activities is an integrated and consistent *system model*, which is expressed in a standardized modeling language (e.g., SysML).

The system model is:

- 1) a set of elements (which represent the system's parts, actors, behaviors, events, requirements, constraints, and test cases),  
and
- 2) the relationships among those elements (e.g., associations, dependencies, generalizations, connectors, flows).

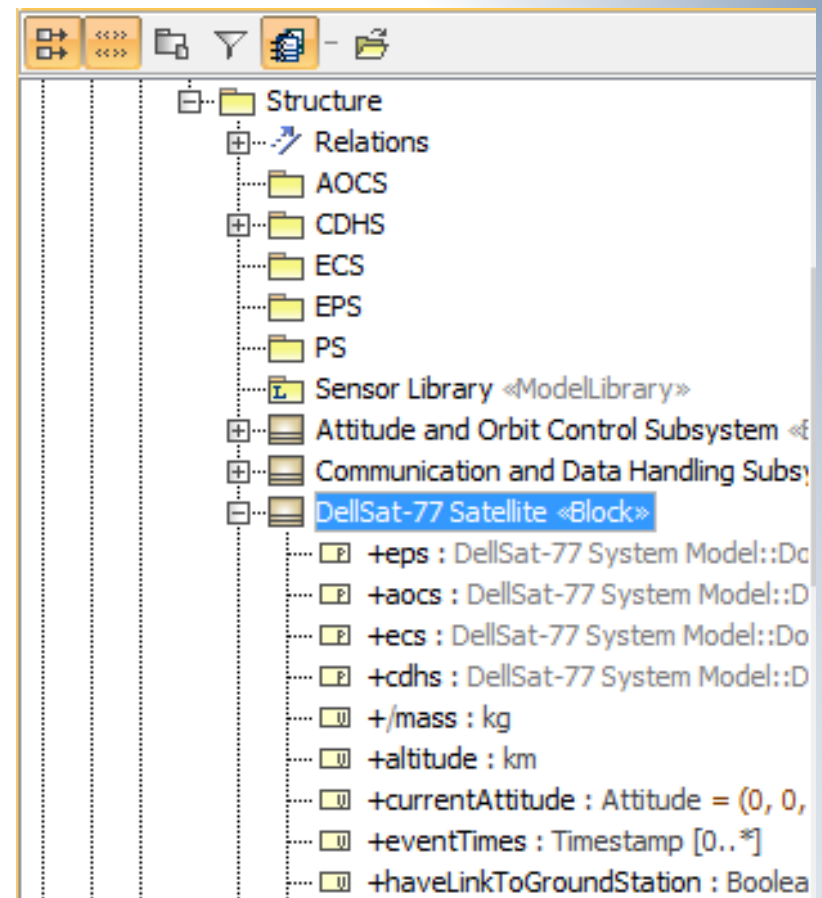


# Overview of MBSE:

## *The System Model*

The system model is created with a dedicated modeling tool and stored in a *model repository* (i.e., a set of data files produced by the tool).

A model user can navigate that model repository in the tool's *model browser* to view (or modify) a specific element and its relationships with other model elements.





# Overview of MBSE:

## *Model-Based Engineering vs. Document-Based Engineering*

With the traditional, document-based engineering approach, designers create a disjoint set of diagrams, spreadsheets, and text documents to convey system design information.

When requirements change or new design decisions are made, a system designer has to know ahead of time which subset of those artifacts is impacted by those changes and where those artifacts are stored. He or she then needs to open each one sequentially and manually make the same change repeatedly in multiple locations—a process which is time consuming and error prone.



# Overview of MBSE:

## *Model-Based Engineering vs. Document-Based Engineering*

With the model-based engineering approach, designers create an integrated system model—an engineering artifact that exists independently of any diagrams that designers may create to provide views of the model for various stakeholders.

When requirements change or new design decisions are made, a system designer makes the necessary changes *one time in one place* within the system model. The modeling tool automatically propagates those changes to all diagrams, spreadsheets, and text documents where the modified elements appear. There is no opportunity for incorrect data entry. Consistency across all views is ensured.



# Overview of MBSE:

## *Root of MBSE's Return On Investment*

The cheapest defect to fix  
is the one you prevent.



# Overview of SysML

SysML is a key enabler for the practice of MBSE. It's a semiformal graphical modeling language used by systems engineers to *visualize* and *communicate* ideas about a system's requirements, structure, behavior, and constraints.

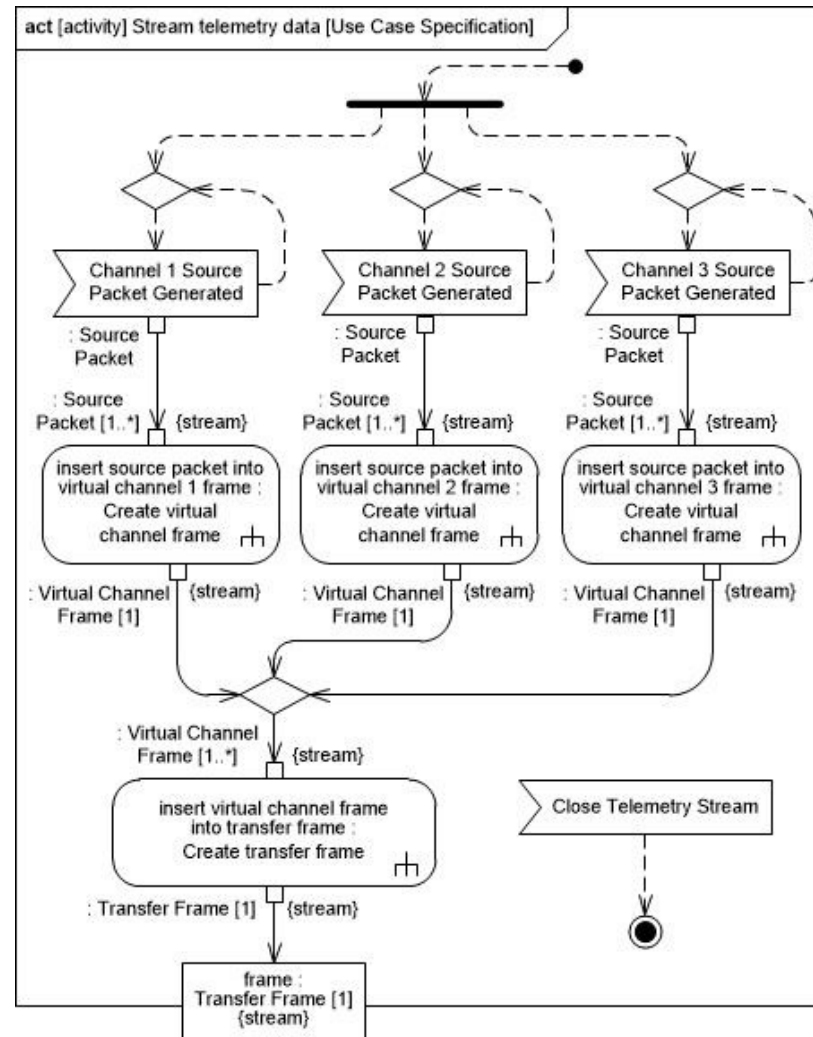
SysML has a grammar and vocabulary—just like any of the natural languages we speak in this world (e.g., Hindi, Japanese, English). Because it's a semiformal language, however, it enables more concise and less ambiguous expression among system stakeholders.





# Overview of SysML:

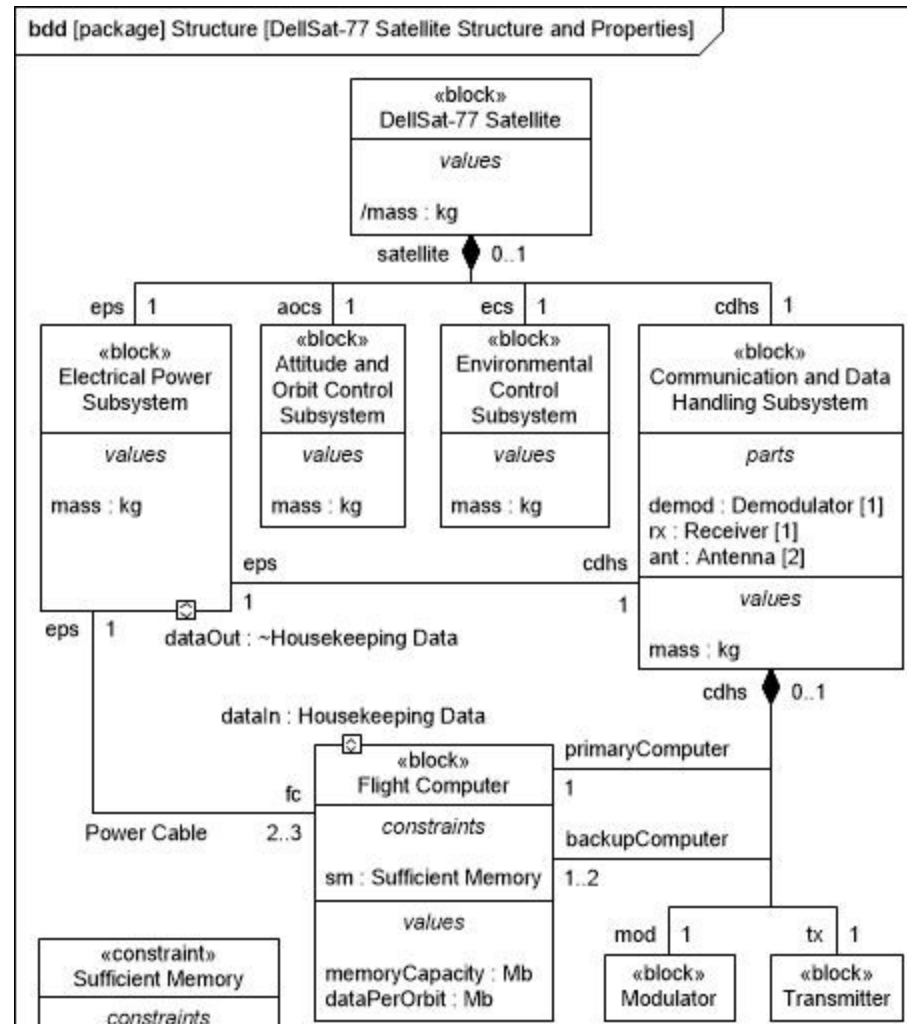
## *A Sample Activity Diagram*





# Overview of SysML:

## *A Sample Block Definition Diagram*





# MBSE ROI:

## *A Case Study*

The set of slides that follow show data collected from four different aerospace and defense programs. Three of the programs used the traditional, document-based engineering approach. One program used the model-based engineering approach.

The names of the programs—and the specific aerospace and defense company—have been omitted for proprietary reasons.

*Note: these slides use the acronym “MBSD” (Model-Based Systems Development). Though the industry has not come to consensus on a set of acronyms, “MBSD” is meant to convey a set of model-based activities that span all engineering domains, while “MBSE” is generally used to refer to model-based activities within the systems engineering domain only.*

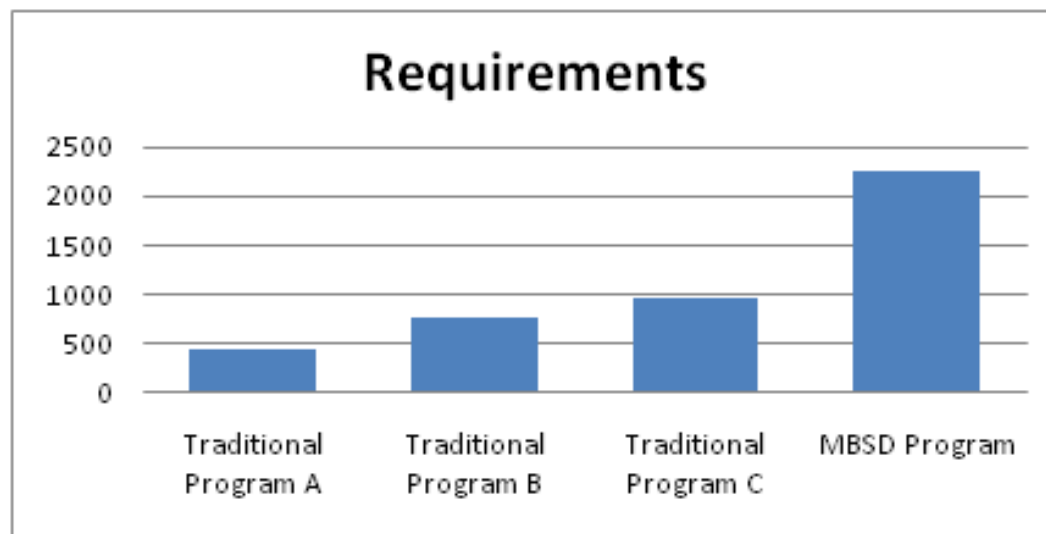


# MBSE ROI:

## *Requirements Comparison*

This figure compares the number of system requirements for these four programs.

The program that used the model-based approach had more than twice as many requirements as the next largest program.



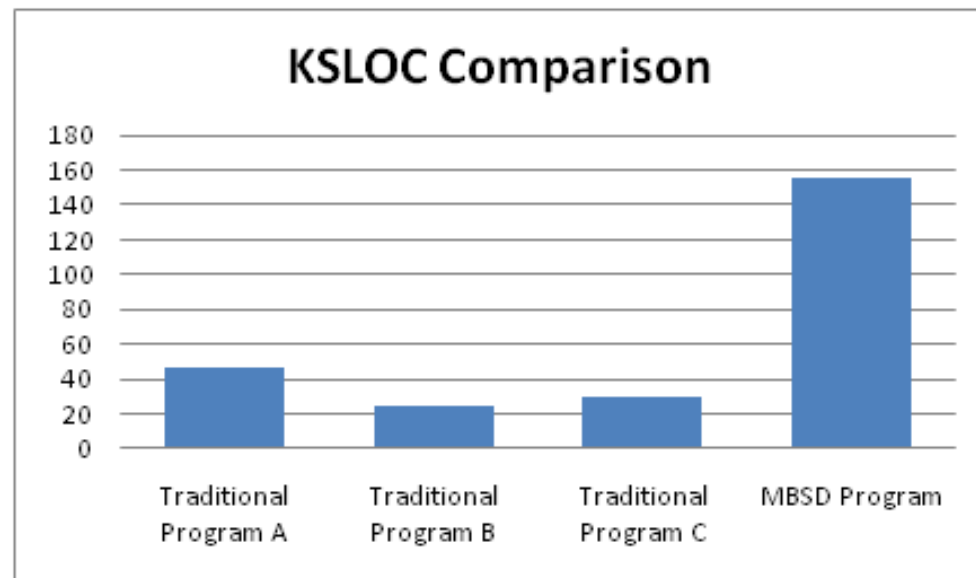


# MBSE ROI:

## *KSLOC Comparison*

This figure compares KSLOC for these four programs as another metric to contrast system size.

The program that used the model-based approach developed a significantly larger system than the other three programs.





# MBSE ROI:

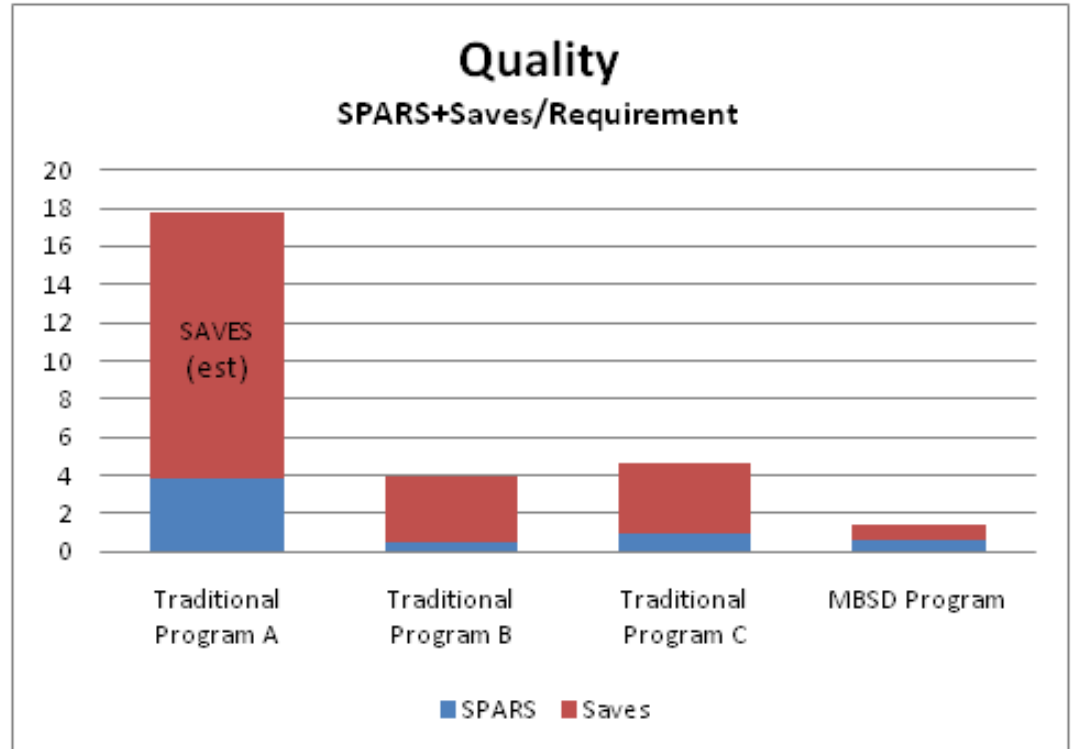
## *Quality Comparison*

This figure compares the number of *defects per requirement* for these four programs.

*Note:*

*“SPAR” refers to a defect caught after release.*

*“Save” refers to a defect caught before release.*



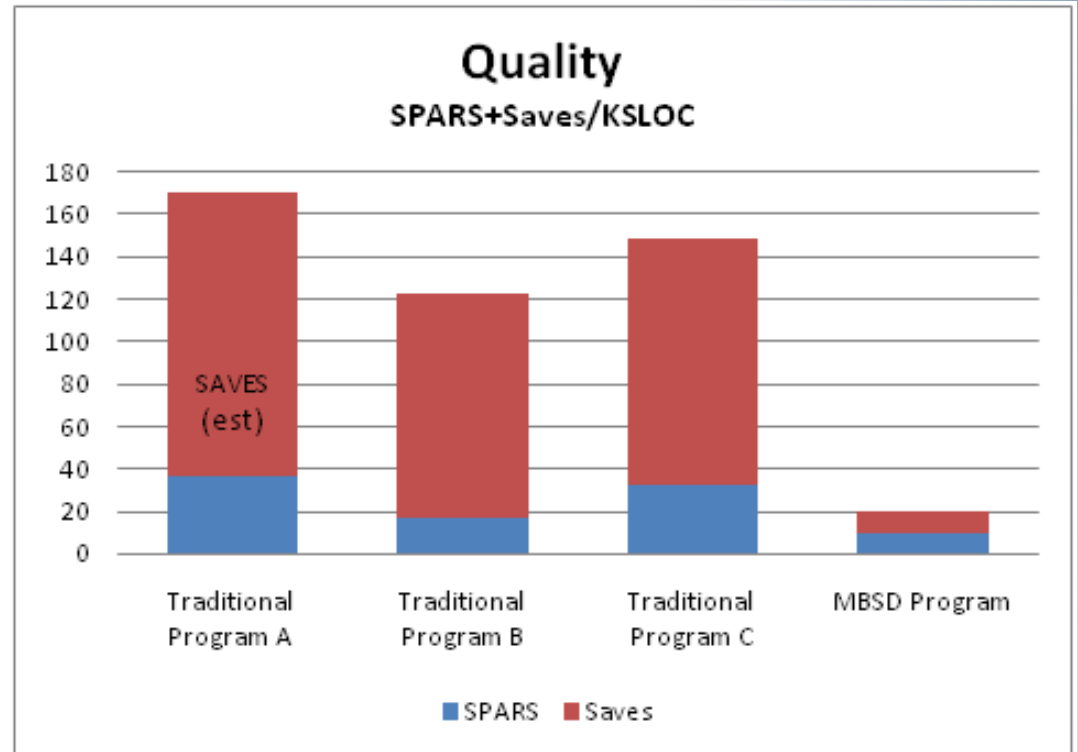


# MBSE ROI:

## *Quality Comparison*

This figure compares the number of *defects per KSLOC* for these four programs.

Though the MBSD-produced system was significantly larger than the other three, its defect density was greatly reduced.



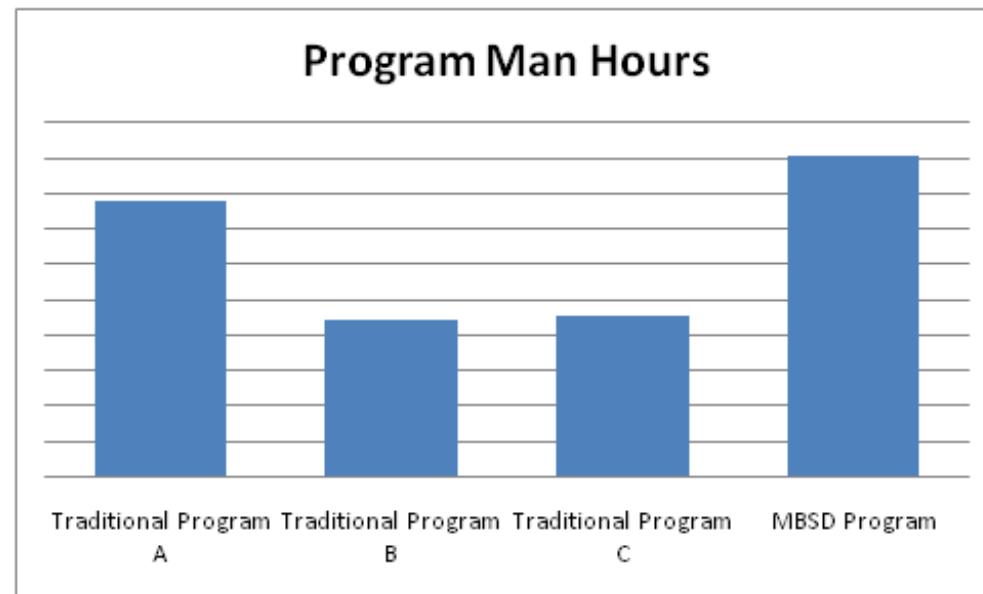


# MBSE ROI:

## *Program Cost in Man-Hours*

This figure compares the relative cost of these four programs in man-hours. (Actual values have been omitted for proprietary reasons.)

As expected, the *total* number of man-hours for the MBSD-produced system was greater than the other three, correlated to system size.





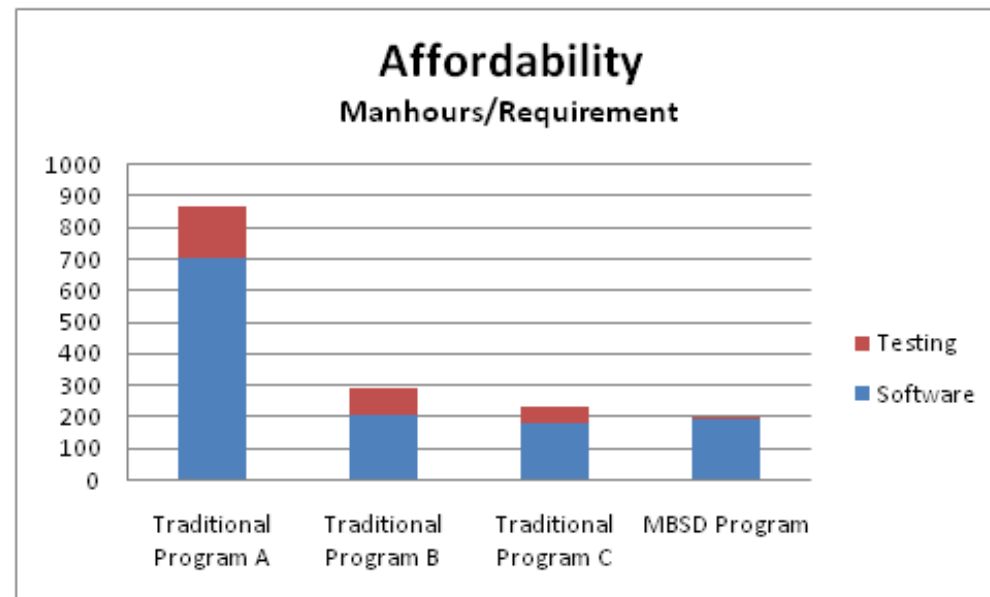


# MBSE ROI:

## *Affordability*

When normalized for system size, however, the MBSD-produced system was developed at a significantly reduced cost.

Program C—the next closest in affordability—was 10% more costly than the MBSD-produced system.



# Contact Information

Lenny Delligatti, M.S., CSEP, OCSMP, OCUP

Phone: (281) 715-0061

E-mail: [ldelligatti@delligattiassociates.com](mailto:ldelligatti@delligattiassociates.com)

Website: [www.delligattiassociates.com](http://www.delligattiassociates.com)



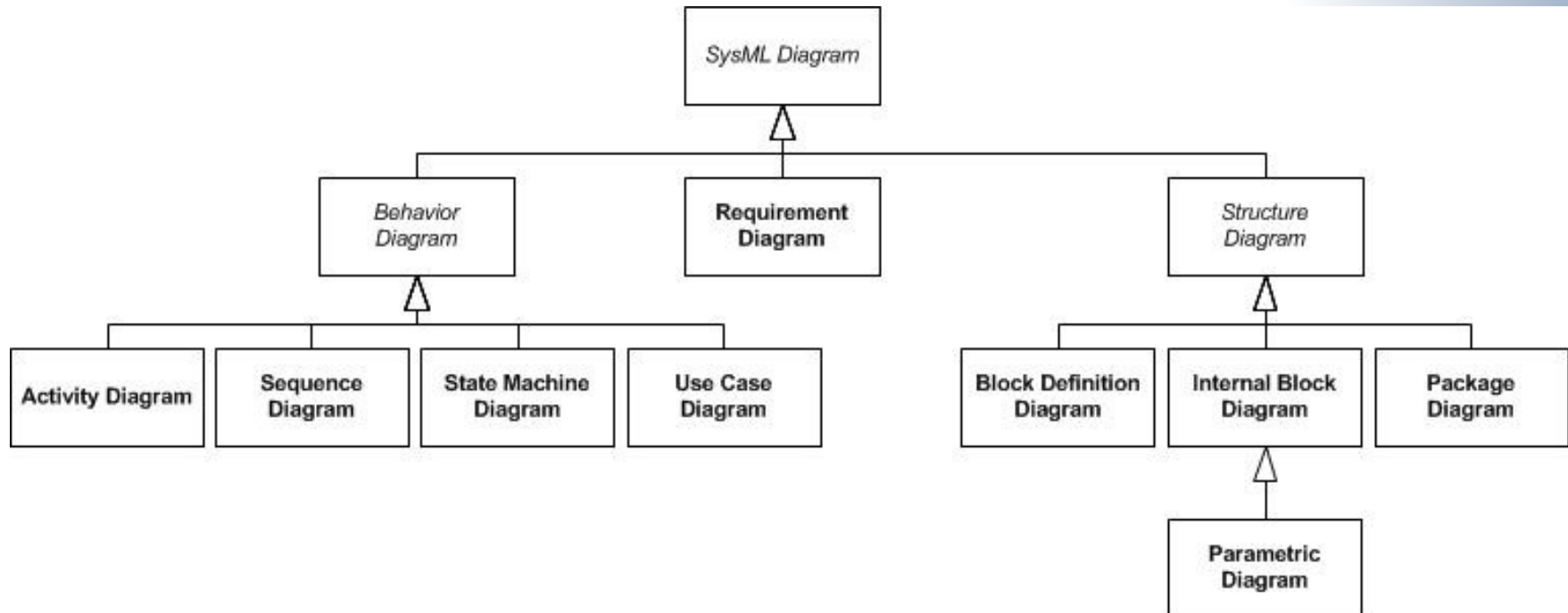


# *Backup Slides*



# Overview of SysML:

## *SysML Diagram Taxonomy*





# Overview of SysML: *Requirement Diagram*

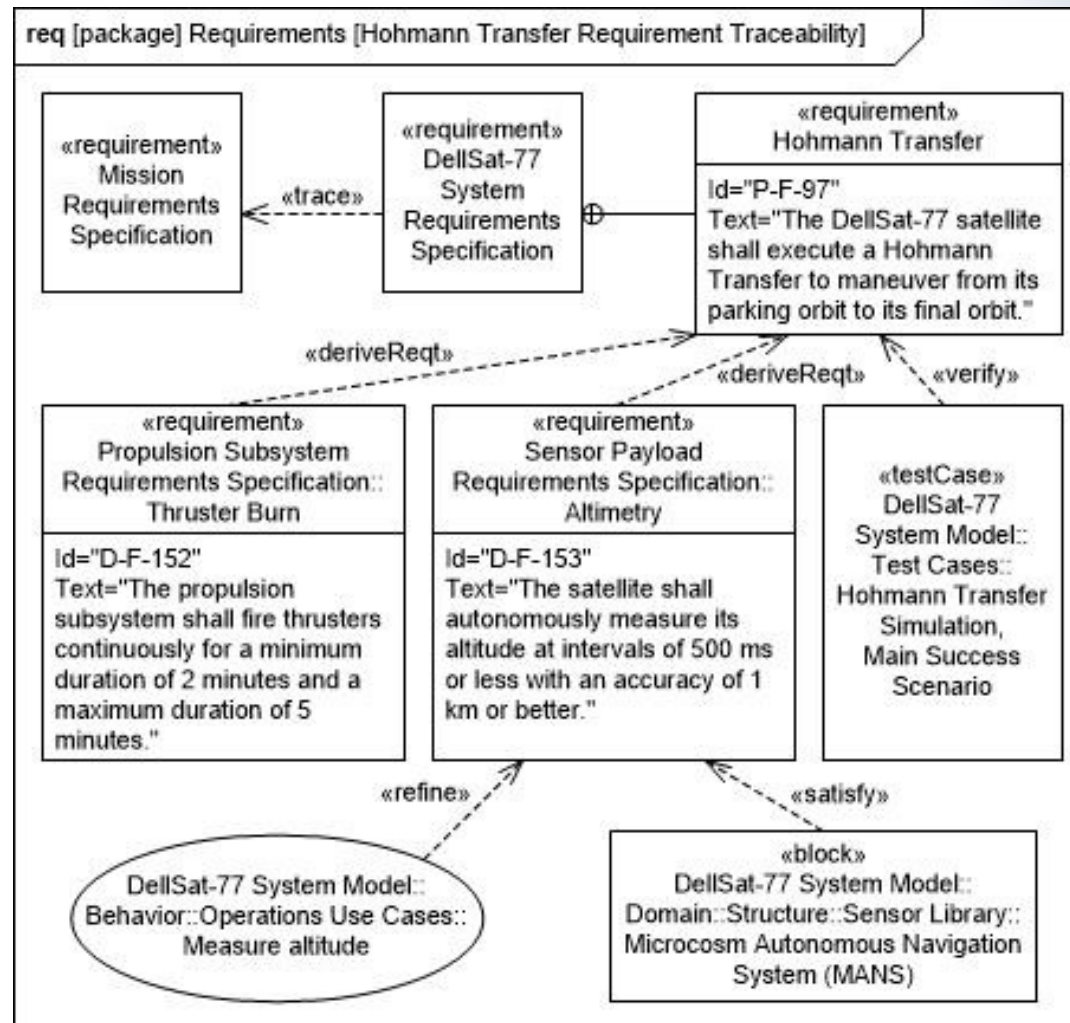
A *Requirement Diagram* is used to display:

- text-based requirements,
- the relationships between requirements (e.g., *containment*, *derivation*, and *copy* relationships), and
- the relationships between requirements and other model elements (e.g., *satisfaction*, *verification*, and *refinement* relationships).



# Overview of SysML:

## *A Sample Requirement Diagram*





# Overview of SysML: *Package Diagram*

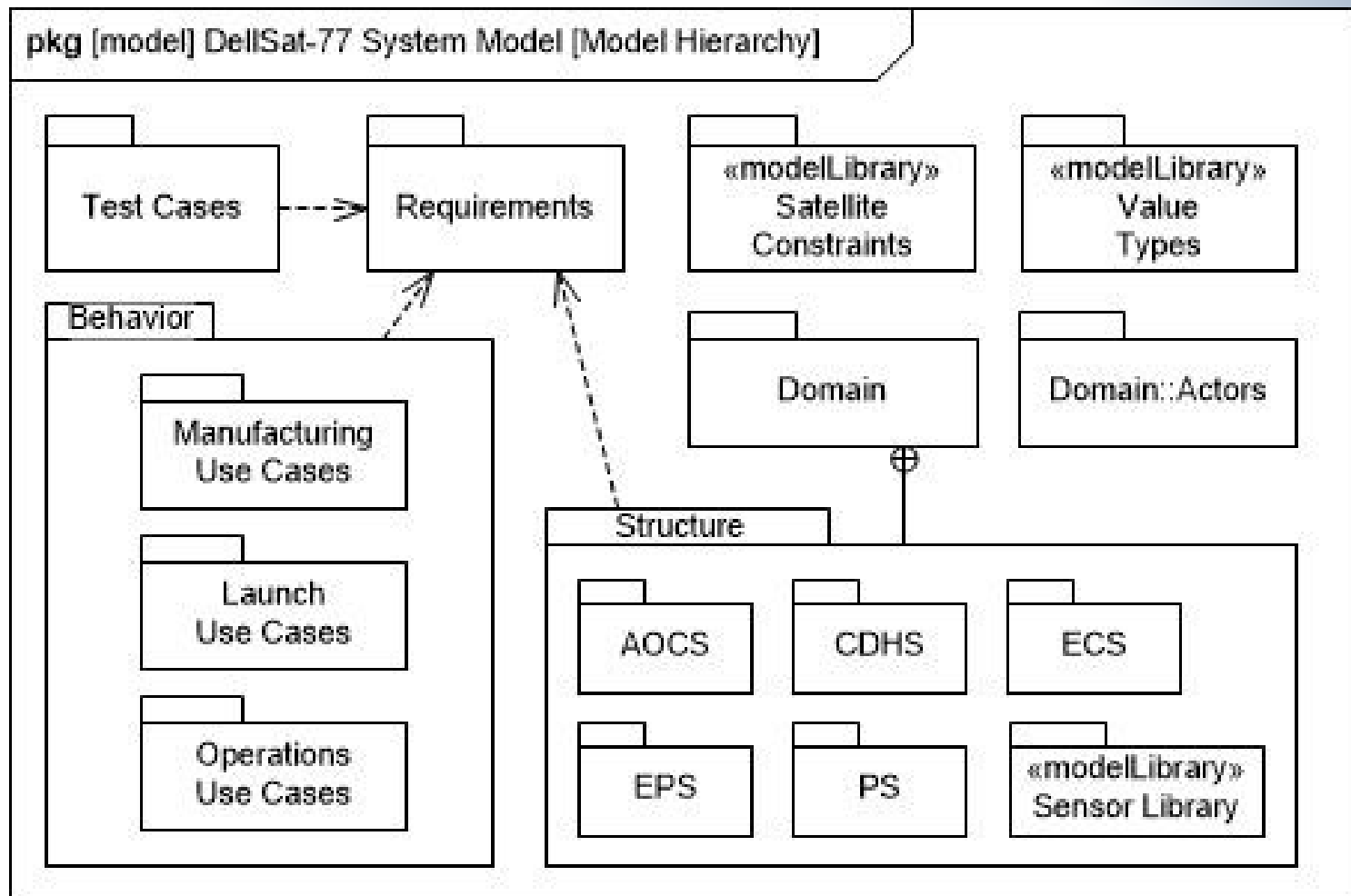
A *Package Diagram* is used to display how a model is organized in the form of a package containment hierarchy.

A package diagram may also show the model elements that packages contain and the dependencies between packages and their contained model elements.



# Overview of SysML:

## *A Sample Package Diagram*







# Overview of SysML:

## *Block Definition Diagram (BDD)*

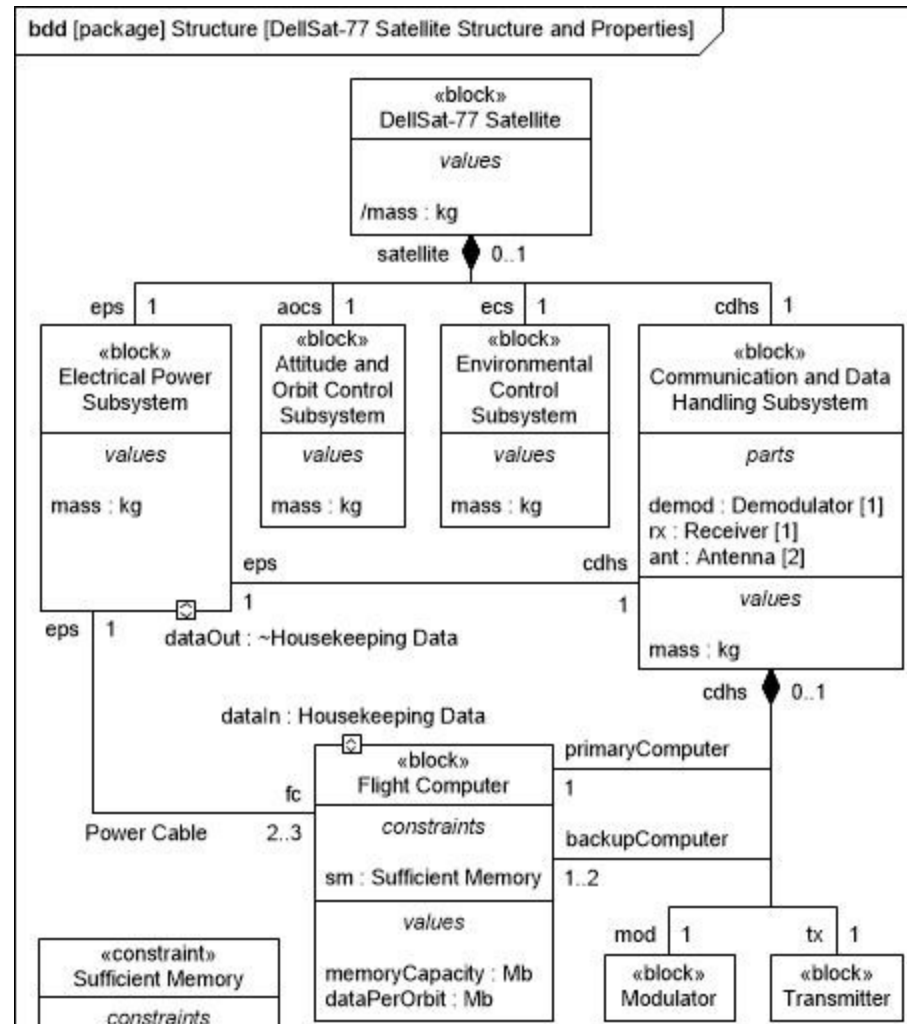
A *Block Definition Diagram (BDD)* is used to display elements of definition (e.g., blocks, value types, constraint blocks, interfaces, flow specifications, actors) and the relationships among them (e.g., associations, generalizations, and dependencies).

Common uses for a BDD include displaying system hierarchy trees and classification trees.



# Overview of SysML:

## *A Sample BDD*





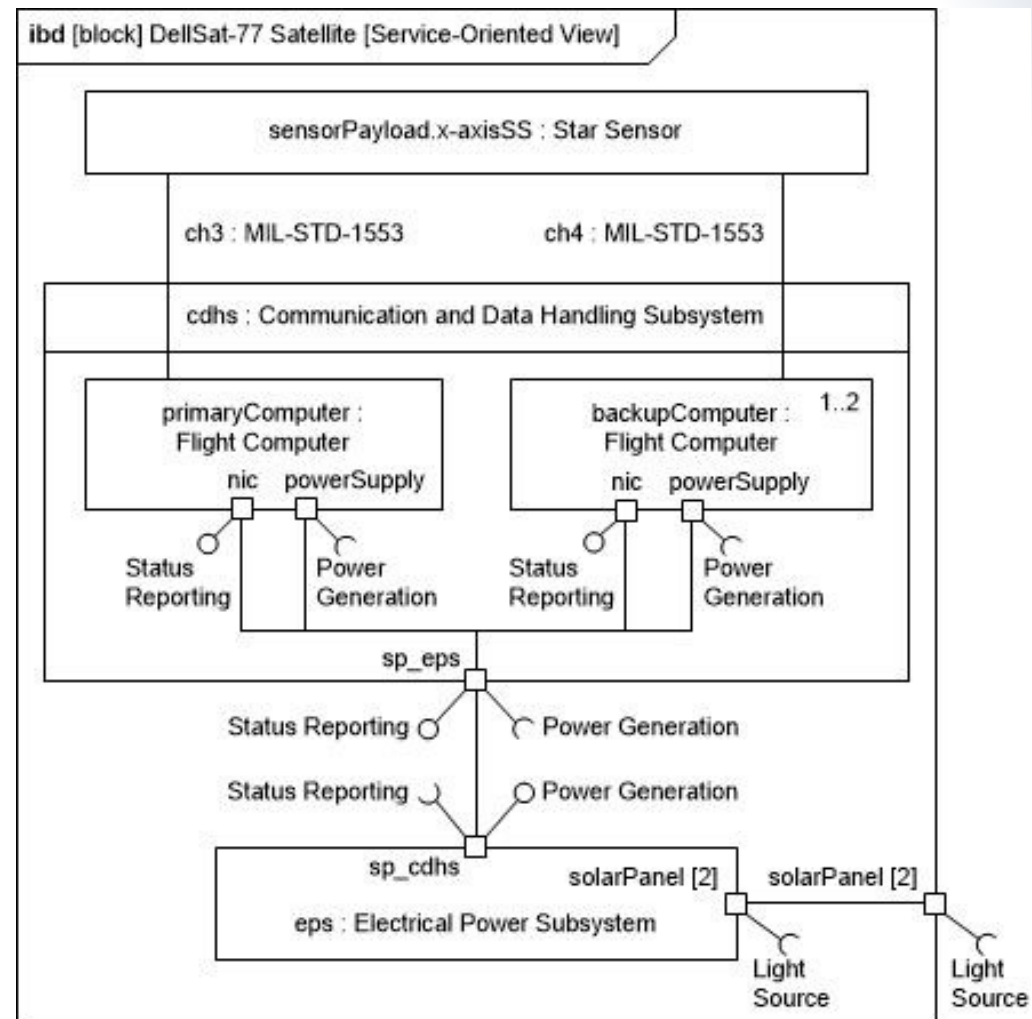
# Overview of SysML: *Internal Block Diagram (IBD)*

An *Internal Block Diagram (IBD)* is used to capture the internal structure of a block (specifically its part properties and reference properties), the connections among those properties, and the flows and services exchanged across those connections.



# Overview of SysML:

## *A Sample IBD*





# Overview of SysML: *Parametric Diagram*

A *Parametric Diagram* is used to express how one or more constraints—expressed as equations and inequalities—are bound to the value properties of a system.

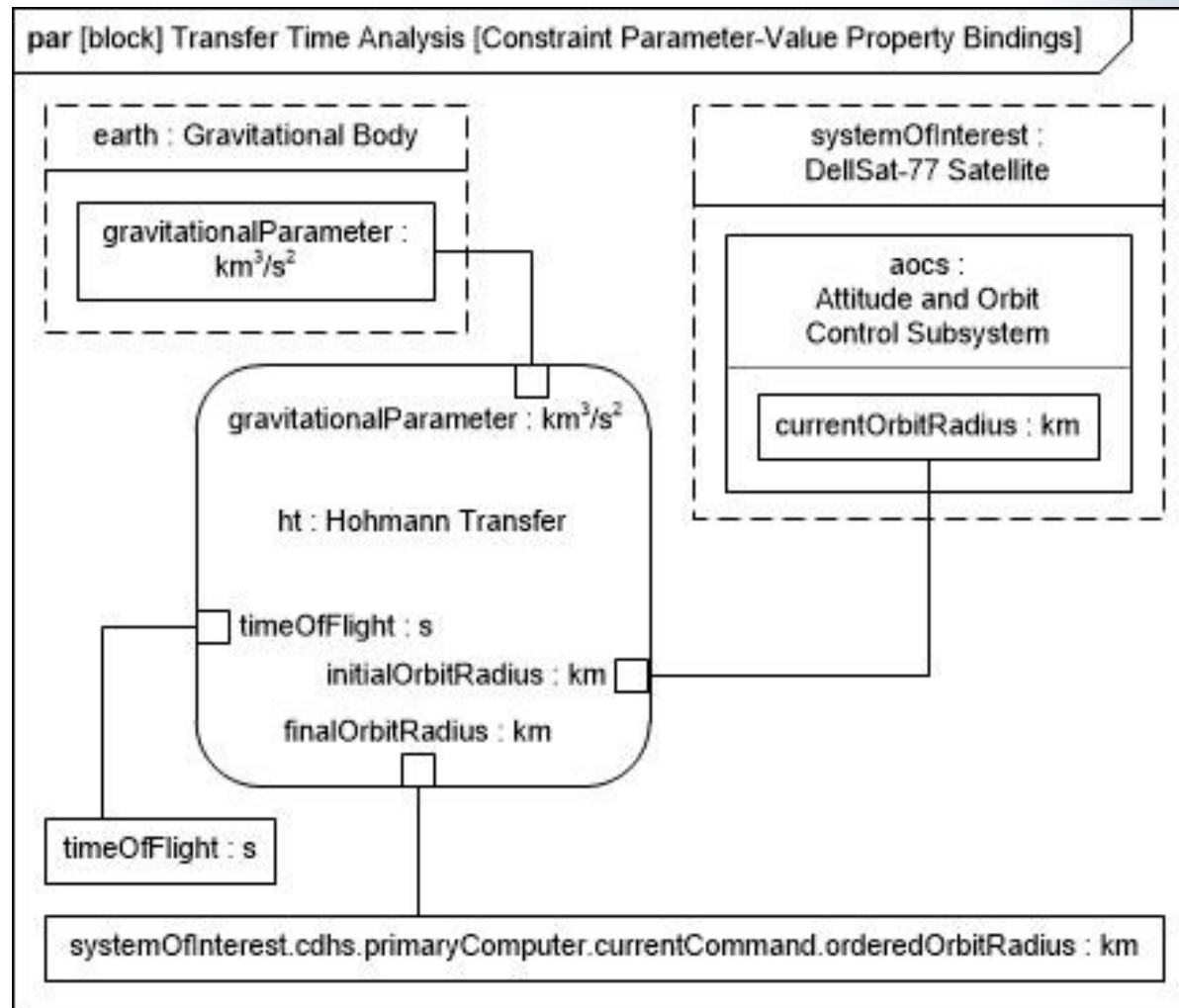
Parametric diagrams support engineering analyses, to include: performance, reliability, availability, power, mass, and cost.

Parametric diagrams can also be used to support trade studies of alternative candidate physical architectures.



# Overview of SysML:

## *A Sample Parametric Diagram*





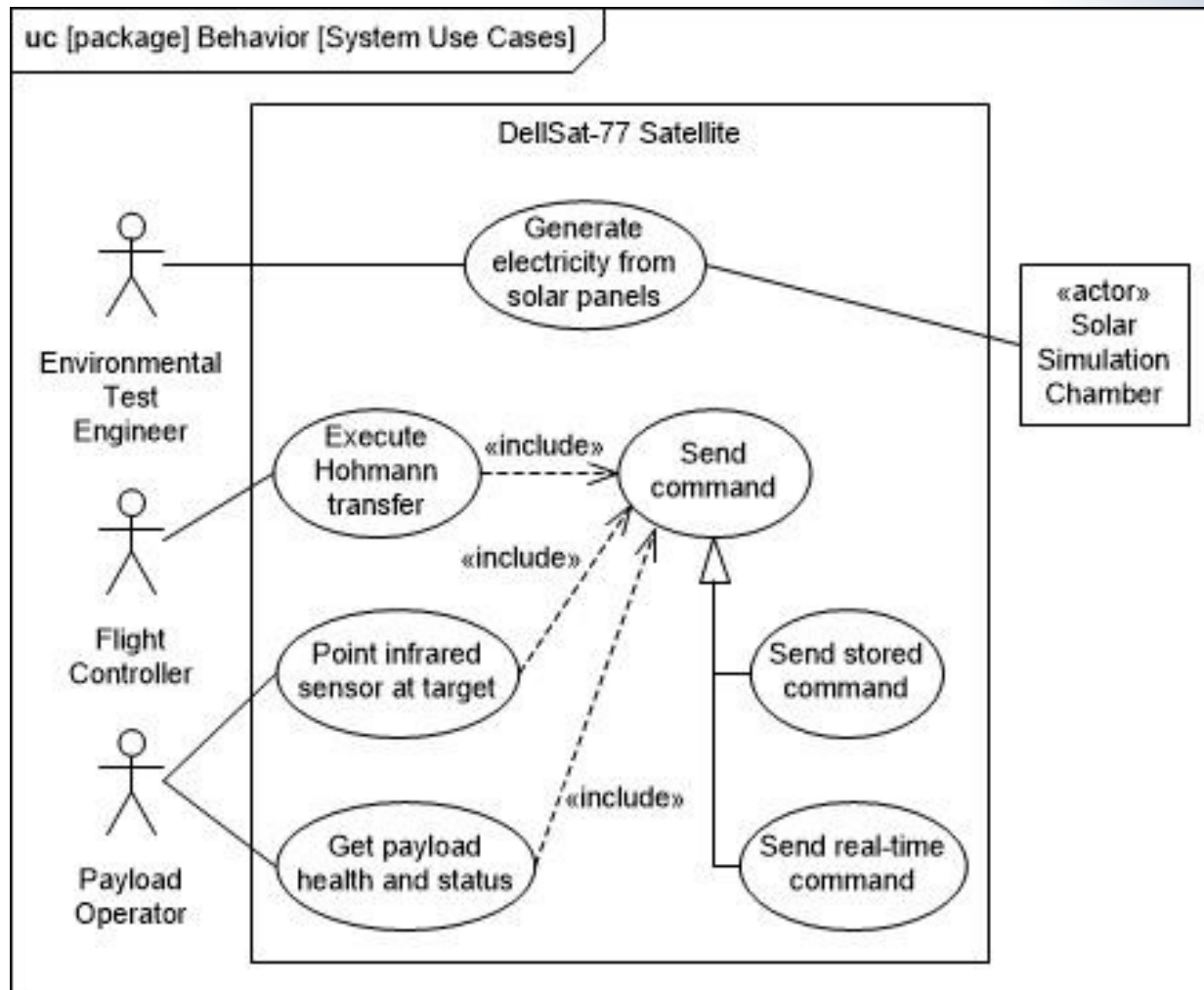
# Overview of SysML: *Use Case Diagram*

A *Use Case Diagram* is used to convey the use cases—the externally visible services—that a system performs and the actors that invoke and participate in those use cases.

A use case diagram is a black-box view of the services that a system performs in collaboration with its actors.



# Overview of SysML: *A Sample Use Case Diagram*







# Overview of SysML:

## *Activity Diagram*

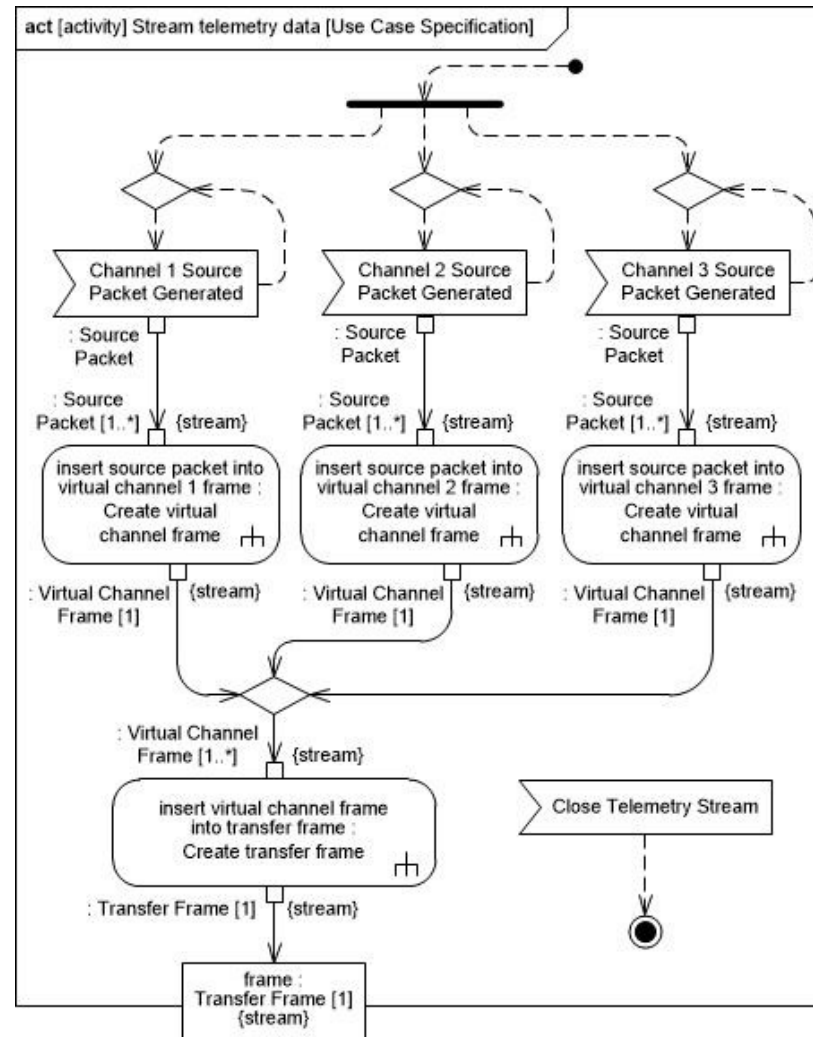
An *Activity Diagram* is used to specify a behavior with a focus on the flow of control and the transformation of inputs into outputs through a controlled sequence of actions.

Activity diagrams are commonly used as an analysis tool to better understand the problem space and to express the desired behavior of the system of interest within that context. It is not intended to be a precise specification of a behavior.



# Overview of SysML:

## *A Sample Activity Diagram*





# Overview of SysML:

## *Sequence Diagram*

A *Sequence Diagram* is used to specify a behavior with a focus on how the parts of a block interact with one another via operation calls and asynchronous signals.

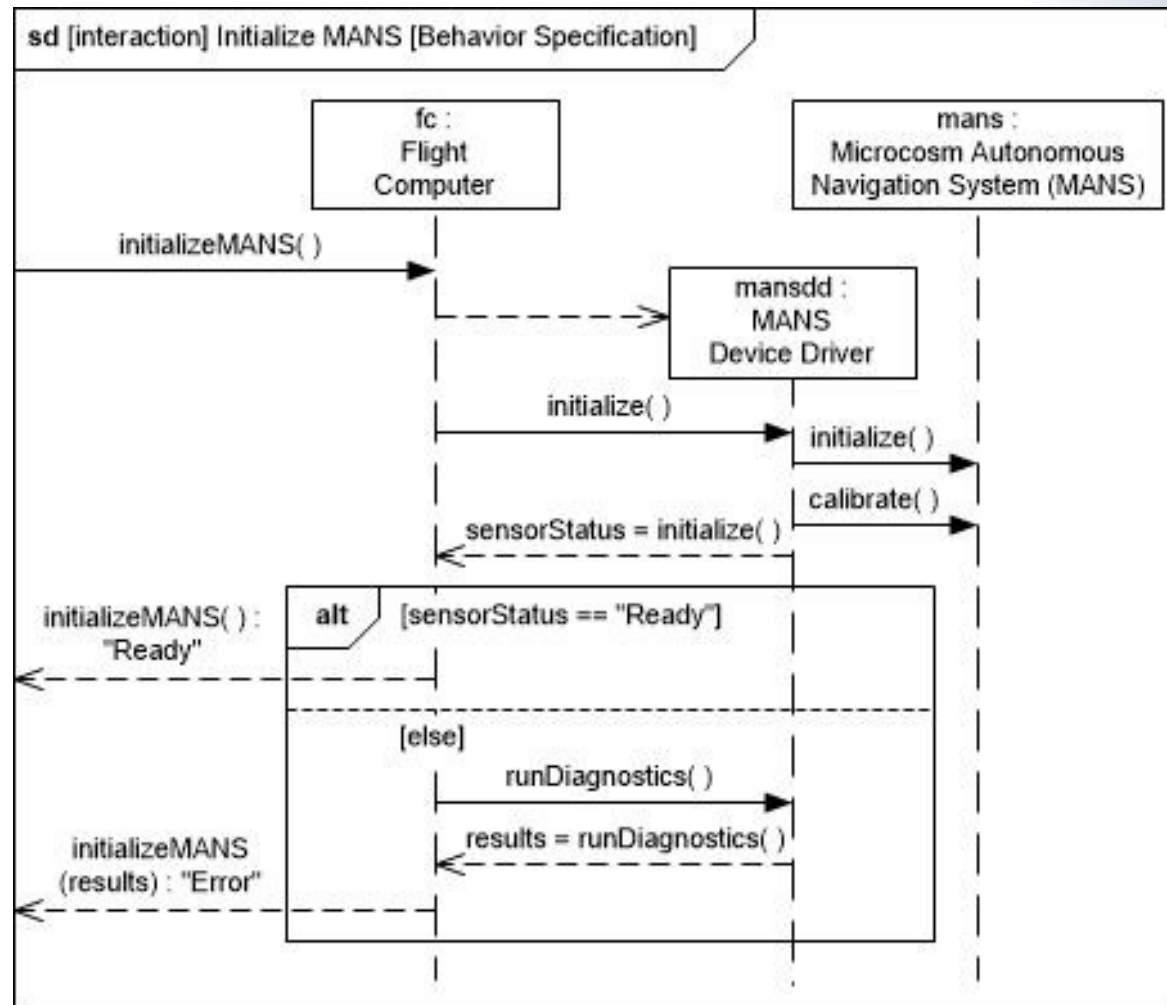
Sequence diagrams are commonly used as a detailed design tool to precisely specify a behavior as an input to the development stage of the lifecycle.

Sequence diagrams are also an excellent mechanism for specifying test cases.



# Overview of SysML:

## *A Sample Sequence Diagram*





# Overview of SysML: *State Machine Diagram*

A *State Machine Diagram* is used to specify a behavior with a focus on the set of states of a block and the possible transitions between those states in response to system events.

A state machine diagram, like a sequence diagram, is a precise specification of a block's behavior that can serve as an input to the development stage of the lifecycle.



# Overview of SysML:

## *A Sample State Machine Diagram*

