Attribute-Oriented Modeling Approach and Its Application to Modeling of Spacecraft Functions

Takahiro Yamada
JAXA/ISAS

Agenda

1. Problems in Model-Based Systems Engineering
2. Attribute-Oriented Modeling
3. Relationships With Other Modeling Approaches
4. An Example – Modeling of Spacecraft Functions
5. Software Tools for Generating Models
6. Conclusion

Problems in Model-Based Systems Engineering

Model-Based Systems Engineering (MBSE)

- Model-Based Systems Engineering (MBSE) uses models as a central and governing role in the specification, design, integration, validation, and operation of a system.
- If models are used, especially formally defined models, in the development of a system, they can
  - Reduce the ambiguity of information that exists in the document-based methodology, and
  - Facilitate sharing of engineering information among different participants involved in the development of the system.

Standard Languages and Tools

- When models of a system are developed, diagrams consisting of boxes and lines are usually used to describe components of the system and how they are connected with each other.
- There are standard languages (like UML or SysML) that specify standard elements and construction rules to draw such diagrams.
- There are also software tools for drawing diagrams according to the specification of UML or SysML.
- Some system engineers have started using these languages, together with software tools, to develop models of the systems that they are developing.
- This certainly reduces ambiguity and facilitates communications among engineers, but there are still problems.

The Problems

- The problems are that,
  - Although UML and SysML specify standard elements for drawing diagrams, it is usually the case that different groups come up with different models of a system even if the same system is modeled and the same language is used.
  - Models developed by a group are viewed by other groups, but they are not processed by software for doing something productive (for example, design, validation, operations, etc.). In other words, models are not used electronically.
- This is because there is no standard that specifies how the components in the systems should be represented with standard model elements.
- Therefore,
  - Each group devises a way to model components.
  - It is not possible to develop software that processes models.

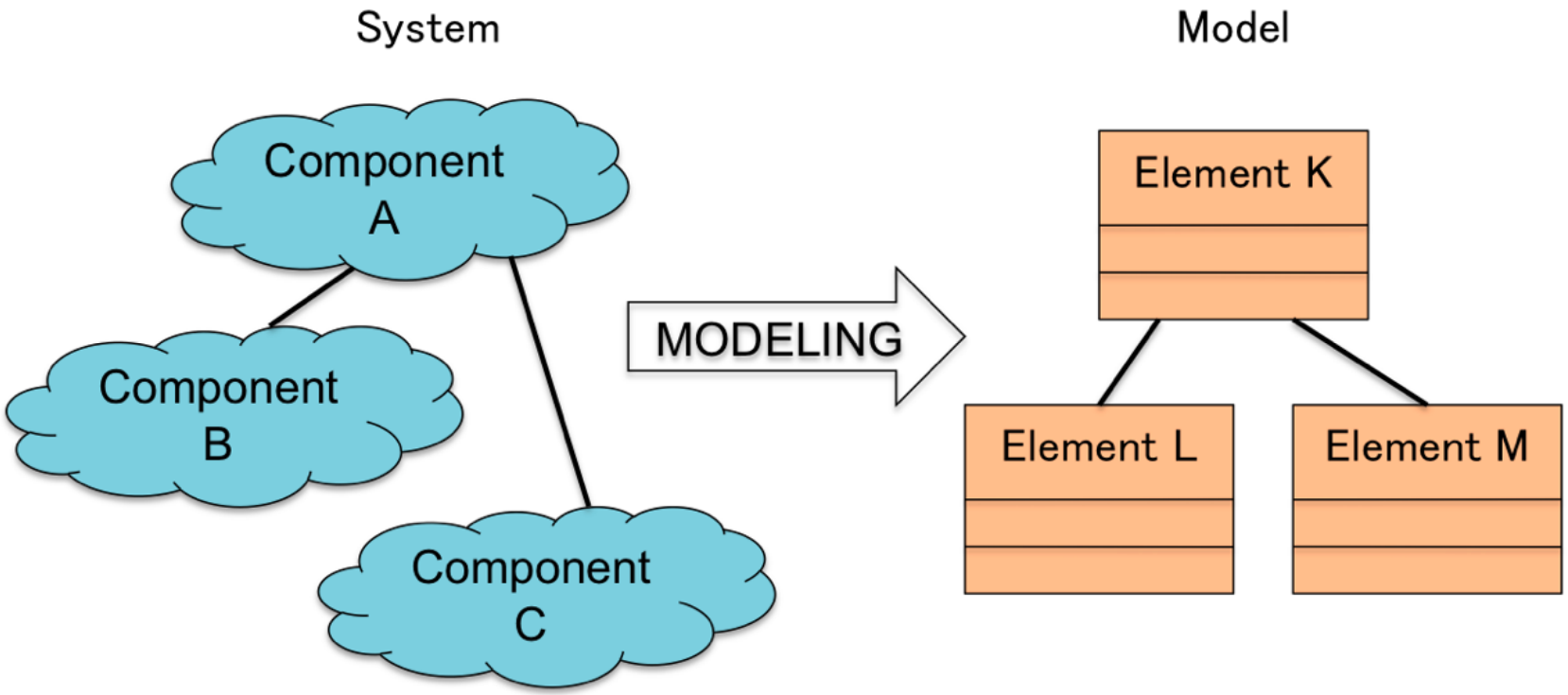
Attribute-Oriented Modeling (AOM)

- In order to solve these problems, this paper proposes a new approach to modeling of systems.
- This approach is called the “Attribute-Oriented Modeling (AOM).”
- The AOM should be used together with a standard modeling language like UML or SysML, not instead of it.
- By using this approach, a system is always modeled in the same way regardless of who models the system.
- Furthermore, by using the fact that systems in a particular domain are always modeled in the same way, we can develop generic software that can process any model in that domain. Therefore, models developed with this approach can always be processed by software.

Attribute-Oriented Modeling (AOM)

What is Modeling?

- Modeling of a system is
  - to represent the components of the system with a set of well-defined model elements.



The Essence of AOM

- The central concept of the AOM approach is that

  a type of model elements used in models in a particular domain

  should be defined with

  a set of attributes that the system components to be modeled by that type of model elements possess.
- In order to do this, we must define types of model elements for each problem domain and a set of standard attributes that each model element type has.
- Types of relationships between model elements should also be defined.
- The set of model element types (together with their attributes) and the relationships among model elements for a particular domain should be standardized and used for every model in that domain.

Example: System of Organizations (1)

- When a system of organizations is to be modeled, we should define a model element type called “organization” to model any organization.
- The model element type “organization” should have attributes like:
  - the name of the organization,
  - its head (person in charge),
  - the number of people in it,
  - its roles,
  - the sub-organizations it owns,
  - etc.
- A relationship between organizations called “owns” should also be defined.

Example: System of Organizations (2)

- Sub-types of “organization,” such as “company,” “school,” and so on, can also be defined.
- These sub-types inherit the attributes that their parent type “organization” has, and they also have their own attributes.
- For example, the sub-type “company” of the model element type “organization” will have attributes like:
  - stakeholders,
  - products,
  - etc.

  in addition to the attributes it inherited from the parent type “organization.”

Development of Models

- A model of a system should be created by representing the system components with model elements.
- In AOM, this is done by specifying, for each component, the values of the attributes that the model element that represents the component has.
- Let's suppose that
  - system component A is to be represented by model element K, and
  - model element K belongs to model element type X.
- Model element type X has its attributes specified.
- Model element K that represents system component A is specified by setting values to these attributes for system component A.

Example: System of Organizations (3)

- Let's suppose that XYZ Company is represented by a model element XYZ_Company of model element type "organization" shown before.
- Model element type "organization" has such attributes as
  - its name,
  - its head,
  - the number of people,
  - its roles, and
  - its sub-organizations.

Example: System of Organizations (4)

- Model element XYZ_Company is specified by setting values to these attributes for XYZ Company.

Model Element Type

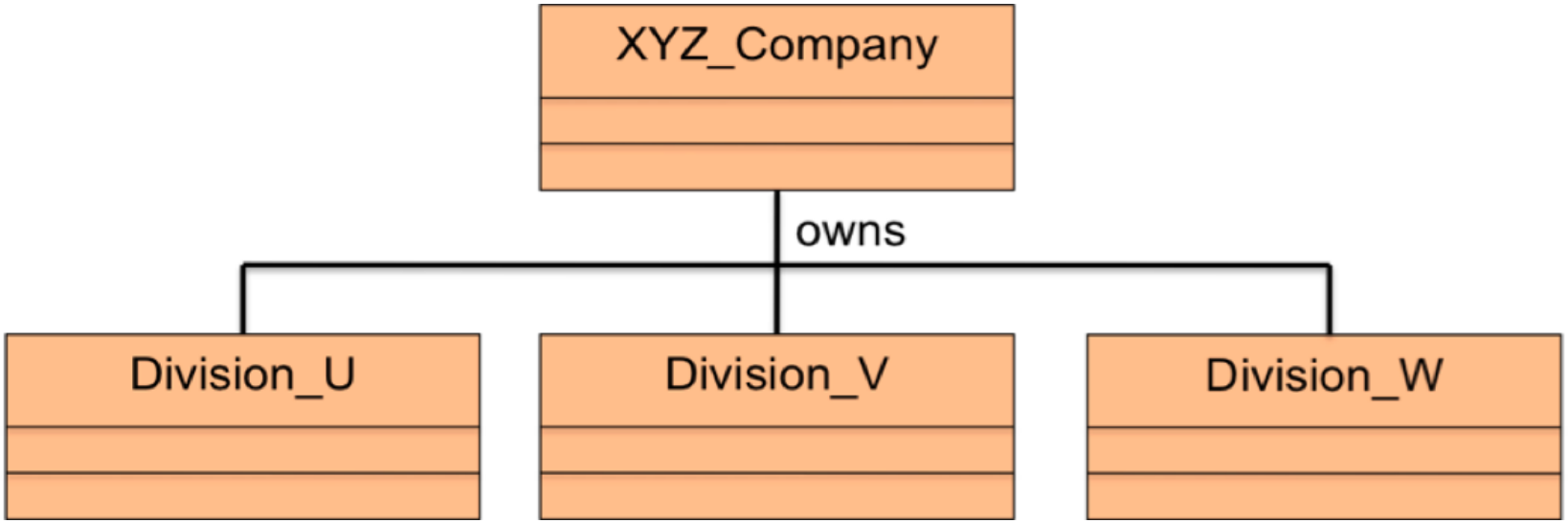
Organization	
Name	
Head	
# of people	
Roles	
Sub-organizations	

Model Element

XYZ_Company	
Name	XYZ Company
Head	Tom Jones
# of people	219
Roles	Manufacturer Consultant
Sub-organizations	Division_U Division_V Division_W

Example: System of Organizations (5)

- The relationship between XYZ Company and its sub-organizations can also be specified by using the relationship “owns” between organizations.



Mapping to Standard Languages

- If model elements are specified for a system in this way, they can be represented with elements defined in UML or SysML.
- How to map model element types and their relationships to UML (or SysML) elements should also be standardized.
- Some people use UML profiles for this purpose, but there are not many UML profiles that specify attributes that model elements should have. Therefore, using a UML profile does not necessarily solve the problems raised in this paper.

Relationships With Other Modeling Approaches

Relationship with UML

- If terms of software engineering are used,
  - model element types correspond to classes, and
  - model elements correspond to instances.
- UML uses these concepts.
- What the AOM approach proposes is, however,
  - to standardize a set of class definitions, each with a set of attribute definitions, to be used in each problem domain.
- These class definitions should be used in every model of that domain in order to
  - Facilitate sharing of models among different groups, and
  - Enable processing of models by software.
- UML provides syntax while AOM provides semantics for models.

Relationship with Ontology

- If an ontology in a problem domain is available, the model element types and their attributes can be defined based on the ontology.
- Languages for defining ontologies (such as OWL and RDF) can be used for defining model element types and their attributes.
- However, how ontologies should be used in defining model element types and their attributes is still a subject for future investigation, and modeling experts and ontology experts should work on this subject hand in hand.

Relationship with IEEE 42010

- Defining model element types and their relationships for a particular domain in AOM is very close to creating a viewpoint defined in the IEEE standard on “Systems and software engineering — architecture description” (IEEE 42010).
- According to this standard,
  - A viewpoint establishes the conventions for the construction, interpretation, and use of models.
- Correspondence:

IEEE 42010	AOM
Defining a viewpoint	Defining a set of model element types and their relationships in a domain
A model	A set of model elements specified based on the definition of model element types

Relationship with RM-ODP and RASDS

- The following standards contain viewpoint definitions:
  - Reference Model of Open Distributed Processing (RM-ODP) by ISO, and
  - Reference Architecture for Space Data Systems (RASDS) by the Consultative Committee for Space Data Systems (CCSDS).
- Each viewpoint definition in these standards contains the definition of basic elements used in that viewpoint, but it does not have definitions of attributes of these basic elements.
- There is an ISO standard that specifies a UML profile for RM-ODP, but it does not specify attributes of basic elements, either.
- These standards should be augmented by defining standard attributes for each of the basic elements.

An Example of Using This Approach for Modeling Spacecraft Functions

Functional Object

- As an example of a model element type of AOM, the concept of the Functional Object is presented.
- To design complex functions performed by a spacecraft, it is a common practice to decompose the functions into groups of functions, each consisting of a small set of functions closely inter-related to each other.
- The Functional Object is a model element type that represents a group of closely inter-related functions.
- What is meant by “functions” here is an abstract representation of the jobs performed by a spacecraft based on instructions given by the operators on the ground.

Attributes of Functional Objects

- A Functional Object has the following attributes:
  - Properties,
  - Operations,
  - Events, and
  - State Charts.

Properties

- A Property is a parameter that represents the status of a certain part or a certain aspect of the Functional Object.
- A Property type is always associated with each Property, which may be integer, real, enumeration, and so on.
- The values of Properties can be monitored by the operators using a communication protocol.

Operations

- An Operation is a function performed by the Functional Object and is invoked by receiving an instruction from the operators with a communication protocol.
- An Operation may have one or more parameters.
- As a result of performing an Operation, the values of one or more Properties usually change.

Events

- An Event is something important that happens inside the Functional Object, and its occurrence may be reported to the operators with a communication protocol.
- An Event may have one or more parameters to explain the details of the Event.

State Charts

- One or more State Charts are used to show the behavior of the Functional Object.
- Each State Chart contains some States and State Transitions.
- A State Transition is triggered either as a result of performing an Operation or by the occurrence of an Event.
- In what State of a State Chart the Functional Object is presently in is indicated by the value of a Property associated with that State Chart. When a State Transition occurs, the value of this Property changes.
- Each State of each State Chart determines the set of basic Operations that can be performed when the Functional Object is in that State. Therefore, the State Charts specify the correct sequences of Operations that can be performed by the Functional Object.

Functional Objects as a Model Element Type

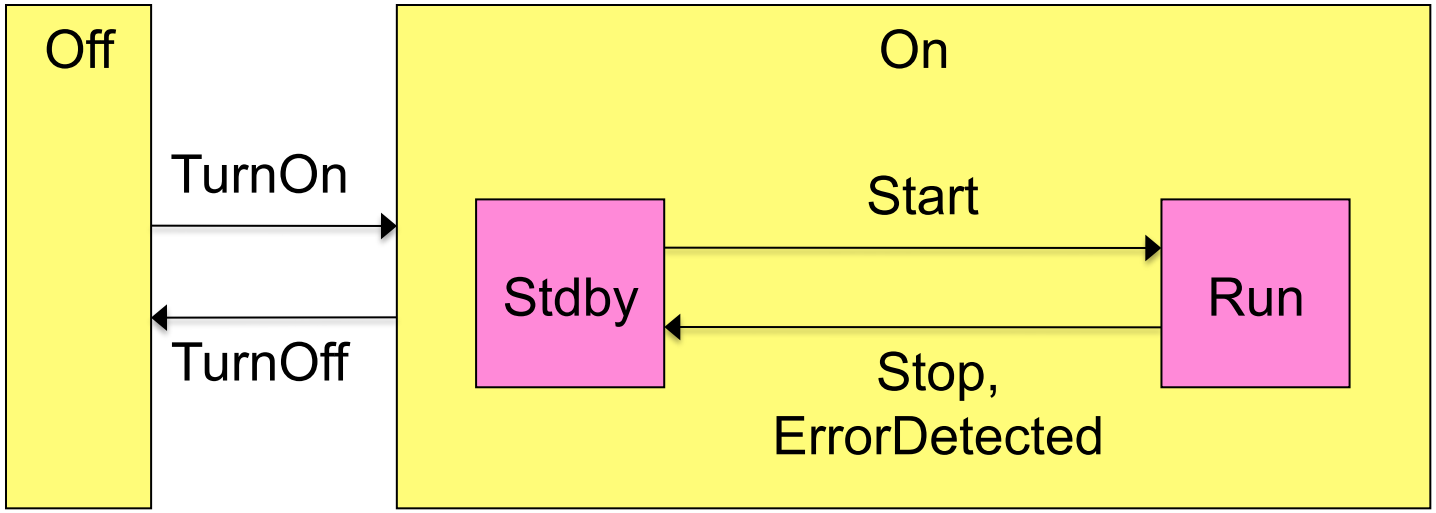
Functional Object	
Properties Property Type	
Operations Parameters Parameter type Property change rules	
Events Parameters Parameter type	
State charts Associated property States Associated property value Performable operations State Transitions Trigger	

Example: Functional Object X (1)

- Properties
  - OnOff (shows whether FO is in On or Off state)
  - RunStdby (shows whether FO is in Run or Standby state)
- Opeations
  - TurnOn
  - TurnOff
  - Start
  - Stop
- Event
  - ErrorDetected

Example: Functional Object X (2)

- State Charts



Functional Object X as a Model Element

Functional Object X	
Properties Property Type	OnOff Enumeration RunStop Enumeration
Operations Parameters Parameter type Property change rules	On Off Start Stop
Events Parameters Parameter type	ErrorDetected ErrorKind Enumeration
State charts Associated property States Associated property value Performable operations State Transitions Trigger	OnOffStates OnOff Off (Off, TurnOn) On (On, TurnOff) OffToOn (TurnOn) OnToOff (TurnOff) RunStopStates ….

Benefits of Using Functional Objects

- Since any Functional Object can be represented in the same way, a uniform model for the functions of spacecraft can be obtained by using Functional Objects.
- The specification of Functional Objects can be stored in a database with a standard format, which can be used by any software for any purpose.
- It is possible to develop generic software that can process any Functional Object because any Functional Object is specified in the same way.

Application

Application of AOM for Developing Spacecraft

- JAXA (the Japanese space agency) has developed models of functions of two spacecraft using this approach.
- The developed models are stored in a standard database and can be used by any software.
- JAXA has developed generic software that can be used for monitoring and controlling any spacecraft. This software uses the fact that the functions of any spacecraft are specified as Functional Objects.
- JAXA has also developed generic software that generates from the definition of Functional Objects a portion of software used in onboard components. This software also uses the fact that the functions of any onboard component are specified as Functional Objects.
- By using this approach, models developed by spacecraft designers can be used electronically for many different purposes.

How Models are Generated and Used

Spacecraft Designers

SIB Generation Tool

SIB Access Tool

Spacecraft Information Base (SIB) (Spacecraft Models Defined as Functional Objects)

An XML document

Generic Monitor And Control Software

Generic Flight Code Generator

Generic Spacecraft Simulator

Future Plan

Generic Test Script Generator

Future Plan

Software Tools for Generating Models

Desired Software Tool

- If a software tool is available for defining model elements based on the definition of model element types, the job of generating models in a particular domain will be facilitated greatly.
- This software tool should read definitions of model element types and let the users fill in values of the attributes for the model elements of their systems.

Existing Software Tools

- There are some software tools that can be used for this purpose.
  - UML tools
    - Have too many features for generating models in a particular domain.
    - Not easy to customize for use in a particular domain.
    - Not a good solution for engineers not familiar with software engineering.
  - XML editors
    - Maybe a good solution for people familiar with XML editors.
  - Excel
    - May be the easiest tool for people not familiar with UML or XML.
    - It is not easy to represent hierarchical data structures.

Ideal Software Tool

- A software tool optimized for generating and manipulating models based on the AOM approach is ideal.
- If there were a tool that can define and manipulate models with graphical user interfaces based on the AOM approach, generating models would be much easier than using the existing software tools.

Conclusion

Conclusion

- In this paper, we have shown the concept of the Attribute-Oriented Modeling (AOM) approach.
- By using this approach, the modelers can specify models of their systems just by setting values to pre-defined set of attributes, and the same models are obtained no matter who generates the model.
- Furthermore, by using the fact that systems in a particular domain are always modeled in the same way, we can develop generic software that can process any model in that domain.
- These features of AOM will
  - Facilitate sharing of models among different groups, and
  - Enable processing of models by software.
- If there were a software tool that can be used to generate and manipulate models based on this approach, generation and use of models would become much easier.