Model-based systems

Revision 1 – 8/7/07 7:11 AM
Revision 2 – 8/8/07 10:24 AM
Revision 3 – 8/17/07 12:36 PM
Accepted for publication in
McGraw Hill 2008 Yearbook of Science & Technology

Harold P. Frisch
Emeritus
NASA/Goddard Space Flight Center

Introduction

Engineers, who design systems using text specification documents, focus their work upon the completed system to meet performance, time and budget goals. Consistency and integrity is difficult to maintain within text documents for a single complex system and more difficult to maintain as several systems are combined into higher-level systems, are maintained over decades, and evolve technically and in performance through updates. This system design approach frequently results in major changes during the system integration and test phase, and in time and budget overruns.

Engineers who build system specification documents within a "model-based systems" environment go a step further and aggregate all of the data. They interrelate all of the data to insure consistency and integrity. After the model is constructed, the various system specification documents are prepared, all from the same database. The consistency and integrity of the model is assured, therefore the consistency and integrity of the various specification documents is insured.

This article attempts to define "model-based systems" relative to such an environment. The intent is to expose the complexity of the enabling problem by outlining "what" is needed, "why" it is needed and "how" needs are being addressed by international standards writing teams.

Technical terminology

Many pre-school children encounter their first systems engineering problem in a story about "The three little pigs". The problem is to build a house. Immediately a cognitive model of a single house forms in the child's mind.

Available construction material is straw, sticks or bricks. This single cognitive model then splits into three alternative solution models. Next the potential of a destructive wind is introduced with a house survival and inhabitant safety requirement. Additionally, the three stakeholders have different degrees of construction cost tolerance and acceptable risk. The models now become clouded with uncertainty. Convergence to a design solution requires learning to gain insight and understanding. A decision-making meeting takes place to evaluate proposals. Each stakeholder attempts to convey their solution model to the others. They differ and none communicate well enough for learning and understanding to drive the respective cognitive models to consensus for decision-making. They leave, proceed to build, and end with two failures and one success.

A model-based system environment would have computer-aided models for construction cost and risk analysis, complemented by small-scale physical or equivalent computer models for alternative house design investigations. These would be executed to evaluate the emergent behavior of different house designs for a range of building material options, expected wind conditions, construction cost and acceptable risk. The model-based systems environment would allow designers to be creative as they propose and evaluate design alternatives, database results, and recall and reuse archived work. Design documentation would be generated from the commonly used database for communication, mutual learning and peer review.

The failure to model leads to a weakened ability to develop the predictive capability called insight and understanding. This weakens the ability to communicate. Weak communication weakens learning and this weakens consensus building.

Systems engineering context

In this generic systems engineering context, cognitive and scaled physical models are foundational to complex systems communication, understanding and decision-making. Only since Newton's contributions have engineers communicated their cognitive models using the language of applied mathematics and only in the last half-century could these equations be represented for digital solution in computer-sensible languages; i.e., languages that computers can read, process and respond to.

The word set "model-based systems" is used freely across domains of study and levels of abstraction. It conveys the direct message that the realized system is based upon a process of modeling, and the implicit message that modeling is mathematical with computer methods providing digital solutions and system data management.

Historical background

Within the engineering community, attempts to realize visions of model-based systems engineering have been ongoing since the 1970's with varying degrees of success. Most attempts encountered problems associated with information interfacing issues. By decade end it became obvious that resolution required internationally approved standards.

On July 11, 1984, the first (International Standards Organization) ISO TC 184/SC4 meeting was held at NIST (National Institute of Standards and Technology). Participating countries included Canada, France, Germany, Switzerland, the United Kingdom, and the United States. The reason for this meeting was to create an international standard that enabled the capture of information comprising a computerized product model in a neutral form without the loss of completeness and integrity, throughout the lifecycle of a product.

Breadth, depth and maturity levels of these and related efforts, in 2007, have the potential of enabling "Model Based Systems Engineering" environments to meet expanding user needs.

Communication of views

Common to all model-based system views is the need to communicate:

- Decomposition,
- Modeling,
- Inference based decision.

Decomposition

Decomposition is the art of viewing a quantity in terms of a combination of "simpler" quantities. Art refers to the ability to select simplifications that are easily learned and communicated; while also providing predictive insights that are robust relative to modest change and uncertainty. Each must also be checkable and have well-defined interfaces so that a model of the whole can be assembled with emergent behavior detected and verified.

Modeling

Modeling is the process of simplification designed to clearly show the basic structure or workings of an object, system, or concept. A model is a representation of a thing, an idea, or a reality that has an execution engine to query the model. The human mind, an electronic computer, a physical test facility, etc., may accomplish the query operation. Results are reproducible and can be verified. They can be presented for communication and documentation in a text language, a graphic language such as a schematic diagram, a computer generated display, etc.

Inference based decision

Inference based decisions and insights are derived by the process of making logical judgments based upon well-known relationships and model results. These must be verified by test with robustness measured to quantify prediction sensitivity to change. The measures provide justification for reuse/redo decisions and form the basis for change messaging directed only to groups doing work that may need to initiate a change response/action.

Underlying model execution and results collection is the need to join and interface data flowing between models. Even for simple systems it is a major effort to insure information consistency as data flows through the networked set of executable models. Complicating the problem further is the absolute need to maintain the consistency of information shared by all engineers and managers across all system life cycle stages.

Support environment

Irrespective of application, the support environment for "model-based systems" must be:

- Learnable,
- Scalable,
- Checkable.

With the need to interface an ever-expanding set of capabilities, users encounter issues related to learning and results checking. Creators and maintainers additionally encounter issues related to scaling. Scaling is fundamentally an interface issue. If interfaces are difficult to work with, new capability cannot be easily added, enabled and checked. Furthermore, unexpected emergent behaviors of the whole can occur; these too must be detectable and checkable. If interfacing difficulties are allowed to perpetuate, unwanted emergent behavior is guaranteed to grow and become progressively more difficult to fix. The ability to scale further ends. Parallel to this is the need to also maintain data base information integrity, consistency, metadata and traceability as usage expands.

Semantic foundations

Resolution demands a computer sensible semantic foundation based upon international standards. This is needed at the:

- Environment specification level,
- Model representation, exchange and sharing level,
- Model management level,
- Model usage level.

Environment specification

At the environment specification level, a semantic foundation for model-based systems engineering exists; created by standard writing expert teams from ISO, (Object Management Group) OMG and (International Council on Systems Engineering) INCOSE. It collects and defines the core functional elements for any systems engineering process. The work provides the ISO-OMG-INCOSE agreed-to basis for work on AP233 (ISO 10303-233 Application protocol: System engineering and design) and SysML (System Modeling Language).

Within the context of systems engineering a "system" must exhibit observable and reproducible properties and have a boundary that separates it from all other things in the environment. It is essential to know what is inside the system and what is outside of it.

Within the same context, "system of systems" can be defined and in like manner "model of models". Realizations of both carry system/model scalability related requirements. These critically important issues relate to interface control, modularity, openness, commonality, rapid upgrade and rapid certification of all component systems/models. These are non-trivial to resolve at both the engineering level and the socioeconomic level where suppliers are protecting market share while customers are seeking a more competitive supplier base.

Representation, exchange and sharing
At the model representation, exchange and sharing level, a semantic foundation is provided by the (Standards for the representation and Exchange of Product data) STEP ISO 10303.

At the highest abstraction level STEP standards consist of Application Protocols (AP's) that service focused domains of study for which software vendors supply competitive products. Each AP collects, in a computer sensible manner, the foundational concepts of a particular domain along with their associated attributes and relationships. Information is made computer sensible by use of the EXPRESS data modeling language. The EXPRESS models of the AP's define neutral read/write data formats for the representation, exchange and sharing of product data information. Software user groups never see or interact with this deep detail; however, they use the STEP enabled standards-based information interfacing capabilities.

Since these neutral formats are standards-based, they provide a time-stable information representation format consistent with the exchange needs of multi-organizational collaboration, long-term data archival and reuse. Neutral formats also provide an access route from outdated software to new versions or best of breed capabilities.

Management
At the model management level there is the need to configuration manage:

- Models and model data associated with system analysis,
- Data associated with system design,
- Version and traceability relationships between design and analysis.

The STEP standards have a "product data management" capability that services these needs. This includes: the metadata of (who, when, where – what, why, how), versioning, context, authorizations, relationships, etc.

Usage (semantics)
At the model usage level there is a need for semantic standards to communicate concepts and system characterization data. At this level of abstraction professional societies and organizational groups establish relevant semantic standards.

Currently there is a gap between what is actually made and what can be made computer sensible. Many groups are addressing the associated problem via Ontology technology via the W3C (World-Wide Web Consortium's) (Web Ontology Language) OWL standards.

OWL can actually be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms; however, it is hard to find references to major projects with a model-based systems engineering environment, at a capability and maturity level, where relationships are captured and used for model based reasoning. The use of OWL, today for taxonomy capture, opens the door for more capability tomorrow. Examples of ongoing work supporting a managed collection of process plant life-cycle data are the EPISTLE Reference Data Library and the controlled natural language Gellish.

Enabling an integrated collection of interoperable models with well-defined interfaces to build a need-based model of models is non-trivial. This is basically an information interface control problem. It can make very effective use of the STEP standards and advances in ontology technology.

Usage (graphical presentation)

Another consideration at the model usage level is model data preparation and presentation. SysML is a graphical modeling language for representing systems and product architectures, as well as their behavior and functionalities. It builds on the experience gained in the software engineering discipline of building software architectures in UML (Unified Modeling Language). SysML has the potential to provide a major contribution to the learnable-checkable problem. It could provide a high degree of consistency across tools in the input data preparation process, the output presentation process and hence in the peer review process of work in progress - by different groups using different SysML based tool sets. Market forces will eventually dictate the degree to which this vision is realized.

URLs:

- www.tc184-sc4.org/SC4_Open/SC4 Legacy Products (2001-08)/STEP_(10303)/
- pdesinc.aticorp.org
- step.nasa.gov/
- www.uspro.org/documents/
- www.ap233.org/ap233-public-information
- www.sysml.org

- www.w3.org/TR/owl-features/

References:

- Sharon J. Kemmerer (Edt) STEP The Grand Experience, NIST Special Publication 939, July 1999; www.nist.gov/msidlibrary/doc/stepbook.pdf
- Michael P. Gallaher, Alan C. O'Connor and Thomas Phelps, "Economic Impact Assessment of the International Standard for the Exchange of Product Data (STEP) in Transportation Equipment Industries," Final Report, Dec 2002. Planning Report 02-5 NIST; www.nist.gov/director/prog-ofc/report02-5.pdf
- J.F. Andary, H.P. Frisch and D.W. Oliver "Model Based Systems Engineering" in A. P. Sage and W. B. Rouse (Eds.) Handbook of Systems Engineering and Management, John Wiley and Sons, Hoboken NJ, 2008.
- J.T. Pollock and R. Hodgson "Adaptive Information", John Wiley and Sons, 2004.