Taking It All In – Representing Multiple Systems of Interest in a Single Model

Stephen Powley, Coventry University, powleys@uni.coventry.ac.uk
Simon Perry, Scarecrow Consultants Ltd., simon.perry@scarecrowconsultants.co.uk

Categorisation

- Accessibility: PRACTITIONER
- Application: RESEARCH
- Theme: Evolving Current Practices
- Topics: MBSE, Architecture

Abstract

The task of identifying a System of Interest (SoI) and its constituent System Elements is central to established Systems Engineering (SE) practice. However, what one stakeholder thinks of as System Elements may be viewed as SoIs by other stakeholders (and vice versa). This is recognised by leading SE reference works, including ISO 15288 and the INCOSE Handbook, but existing Model-Based Systems Engineering (MBSE) techniques fall short when it comes to representing multiple points of view in the same model.

This shortcoming is one reason why it is often difficult to share model parts effectively within organisations or with suppliers. The result is that multiple models containing their own representations of the same SoI must be created and maintained. This increases the maintenance burden, can lead to inconsistencies, and reduces the ability to understand and control emergent properties.

These issues are addressed in this paper by proposing a simple underpinning ontology and a modelling technique for capturing multiple system perspectives. The ontology is compatible with leading SE reference works and the technique uses standard Systems Modeling Language (SysML) notation. Using standard SysML ensures broad applicability, due to its wide adoption, and means that the technique can be implemented in any tool that supports SysML.

The proposed technique marks model elements with multiple stereotypes (e.g. both «system» and «system element»). A tag associated with each of these stereotypes then contains a reference to the context in which that element is considered a «system» or «system element». The inclusion of context references on elements when they are used in a diagram allows anyone using that diagram to see the contexts associated with the stereotypes.

Broader applications for this technique beyond hierarchical structures are also discussed, including the potential to support multiple, possibly conflicting, ontologies within the same model.

Introduction

The task of identifying a System of Interest (SoI) and its constituent System Elements is central to established Systems Engineering (SE) practice. Achieving this requires the SE team to account for multiple points of view and the related methods and processes are cornerstones of key reference works in the field. There are two prominent approaches, which consider breadth and depth of analysis respectively:

1. Separation of concerns [Dijkstra 1982] relating to different aspects of a single SoI whose architecture is under consideration. Here, the architecture of a system constitutes "*what is essential about that system considered in relation to its environment*" [ISO 2011], whereby a single environment context is considered.
2. Treating the system as a hierarchical set of holons[1] [Koestler 1967], that is multiple SoIs at different levels of granularity. Here, multiple elements of the top-level system are considered as SoIs in their own right in different environment contexts [ISO 2015] [INCOSE 2015].

Approach 1 is already well supported by modelling techniques that help systems engineers express multiple conceptualizations of the same SoI. ISO 42010 [ISO 2011] *"Systems and software engineering — Architecture description*" is a widely adopted international standard that defines an approach "*to produce one architecture description expressing one architecture for one system-of-interest*." The key modelling languages used in systems engineering such as UML [omg.org 2017b], SysML [omg.org 2017a] and OPM [Dori 2002] are also structured around the core idea of separating structural and behavioural concerns.

Approach 2 is discussed in ISO 15288 [ISO 2015], which identifies hierarchical system structure as a "*Key concept*" and early on introduces the idea that *"For more complex systems-of-interest, a prospective system element may itself need to be considered as a system (that in turn is comprised of system elements) before a complete set of system elements can be defined with confidence.*"

The key distinction in Approach 2 is that the System of Interest is considered in different contexts where stakeholders need different levels of detail. These contexts could be, for example: a system assembler and the independent developers of sub-systems; or the design and test departments of the same company.

The focus here is on modelling holons within a single, coherent model (Approach 2), but the technique can also be applied to enhance Approach 1.

The Problem

Both approaches discussed above consider that a top-level System can be decomposed into multiple System Elements [ISO 2015], but Approach 1 assumes that different levels of detail of the same System Element are captured in separate models. Working with separate models has some advantages, such as separation of concerns, more granular configuration management and ease of model control. However, separate models also have significant disadvantages. The holons in the system breakdown structure often reflect the business units that are responsible for each SoI, in an example of Conway's Law [Conway 1968]. This precludes the possibility of constructing viewpoints that address concerns spanning the System Element hierarchy, such as safety and security. The lack of visibility across abstraction layers also impacts the system engineer's ability to develop functions that take advantage

[1] Per Koestler [1967], holons are "intermediary structures on a series of levels…: sub-wholes which display, according to the way you look at them, some of the characteristics commonly attributed to wholes and some of the characteristics commonly attributed to parts. … [The term is derived] from the Greek holos = whole, with the suffix on which, as in proton or neutron, suggests a particle or part."

of desirable emergent properties. It also makes it difficult to construct model queries spanning the complete system that could deliver valuable business intelligence.

Although some Model-Based Systems Engineering (MBSE) tools may support referenced models as black boxes, in the authors' experience existing MBSE techniques fall short when it comes to representing multiple SoIs with full granularity in the same model. This shortcoming can prevent organisations from sharing model parts with internal and external suppliers effectively and can lead to multiple models containing their own representations of the same System Elements. This increases the maintenance burden and can lead to inconsistencies.

The central problem is that currently models generally only capture systems from one point of view (context) in the sense of Approach 2. In this paper we therefore propose a method to address the lack of modelling techniques available to Systems Engineers that allow them to model holons, that is to show how one stakeholder's System is another stakeholder's Sub-system.

A Solution

The issues outlined in the Introduction are addressed here by proposing a modelling technique for describing holons in a single model. The technique uses standard SysML notation, which has broad applicability in SE and is widely adopted. It also applies to UML (which underpins SysML).

The technique is illustrated by building on a simple ontology, which is broadly compatible with leading SE reference works [ISO 2015] [INCOSE 2015] [ODUSD 2008] and based on the system structure analysis and the Framework for Architecture Frameworks presented in [Holt & Perry 2018].

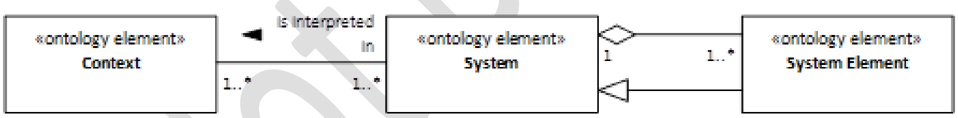

Figure 1 – Simple System Ontology

A System is made up of one or more System Elements, which are themselves types of System. Both Systems and System Elements are interpreted in a Context.

This diagram is a deliberate simplification to illustrate the technique. Thus, interactions between Systems and between System Elements have been omitted, as has the possibility for System Elements to be parts of multiple Systems. The key point is that we have a hierarchical structure[2], with each level interpreted in a Context.

A SysML profile can then be developed from such an ontology. The exact method of profile definition is tool dependent, but an example diagram used to generate a profile is shown in Figure 2.

[2] Because it is a type of System, System Element can also aggregate other System Elements.

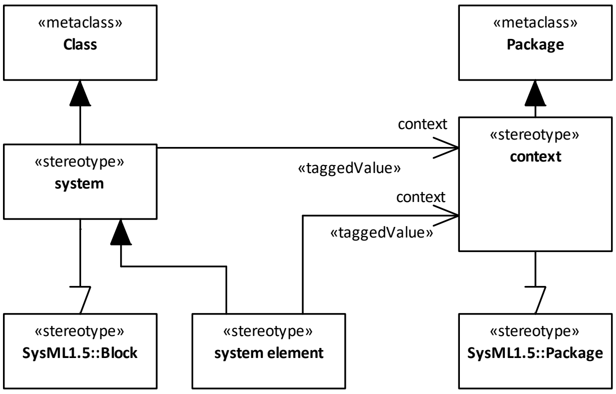

Figure 2: Profile for System concepts with context tagged values on stereotypes

In such a profile, each Ontology Element (System, System Element and Context in this example) becomes a stereotyped SysML block. Rather than representing the 'is interpreted in' relationship of Figure 1 as a stereotyped relationship (e.g. a stereotyped dependency), the link is captured as a 'context' tagged value associated with each stereotype. The 'context' tag can then be used to store a reference to the Context in which that element is considered a «system» or «system element». In this example Context is captured by grouping diagrams for a particular context in the same SysML Package.

To complete our technique, we make use of the support in SysML (and UML) for multiple stereotypes. Multiple stereotypes on a single element can quickly become very confusing. A profile of the type shown in Figure 2 allows the use of stereotype compartments or call-out notes on a diagram (which may be automatically populated, depending on tool support) that contain the 'context' tagged value for each stereotype. This allows anyone using the diagram to clearly see the Context associated with each stereotype (see Figure 4). This ability to show Context explicitly associated with the applicable stereotype was the reason for choosing to use tagged values rather than some type of stereotyped relationship.

Example

Engineering companies do not usually describe systems using theoretical terms like holon, or in simple hierarchies of System and System Element. Rather, they assign specific names to different levels of abstraction in a System of Interest.

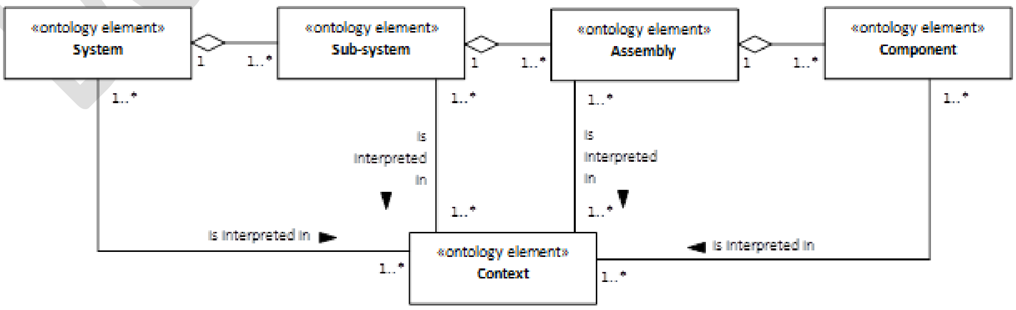

Figure 3 – Typical system hierarchy

Figure 3 shows a typical system hierarchy, with a System made up of one or more Sub-systems, which in turn are made up of one or more Assemblies. Each Assembly is made up of one or more Components. Systems, Sub-systems, Assemblies and Components are all interpreted in a Context.

Note that each instance of a System or its subordinate holons will only be 'interpreted in' a single Context if it has a single stereotype (Approach 1), but assigning multiple stereotypes allows the same instance to be represented as a holon i.e. in multiple Contexts. This diagram (an Ontology Definition View [Holt & Perry 2018]) would be used as the basis for a tool profile in a SysML modelling tool, in a similar way to the example described in the A Solution section. Each of the Ontology Elements shown would become a SysML stereotype and the 'is interpreted in' relationships are realised as 'context' tags associated with the stereotypes. In this case, the single «stereotype» block for System Element of Figure 2 would be replaced by one for each of Sub-system, Assembly and Component. Each stereotype block would have a 'context' «tagged value» that references the 'context' stereotype block.

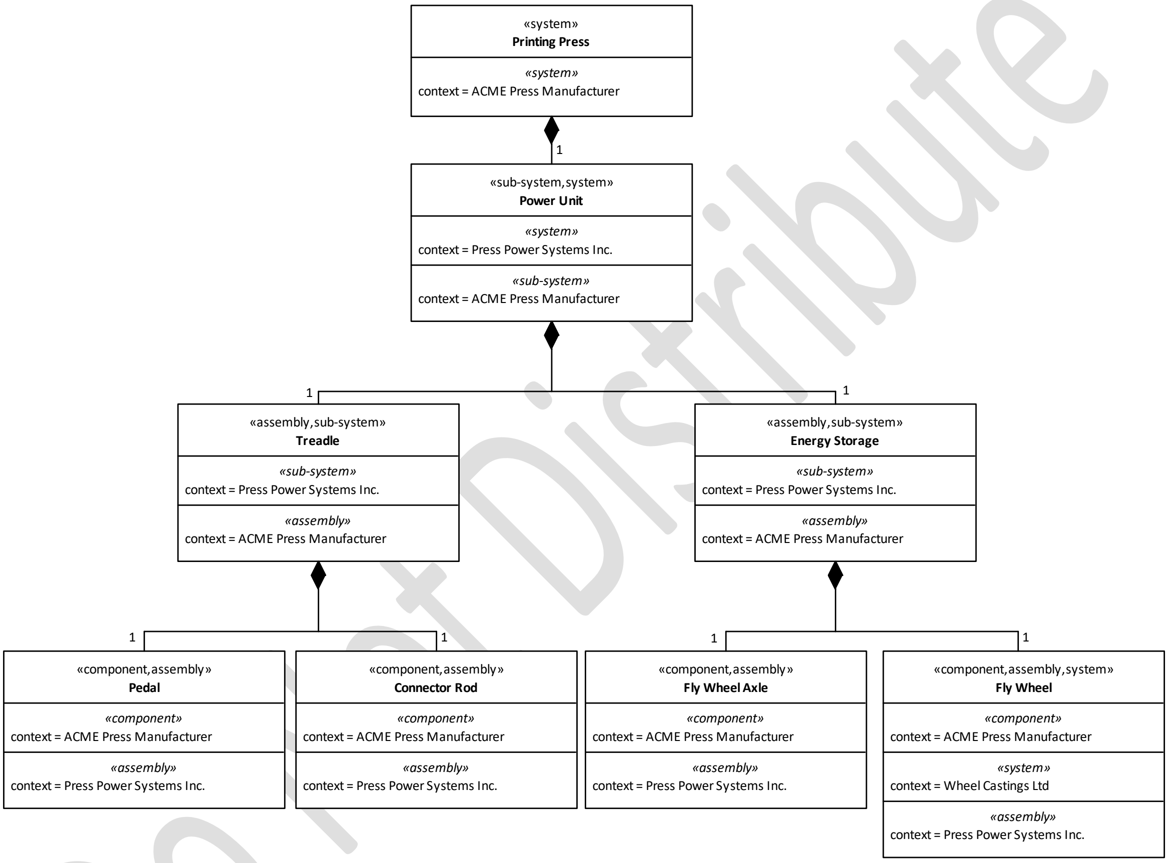

Figure 4 – Specific example of multiple stereotype use

Figure 4 shows an example of the use of multiple stereotypes with associated 'context' tags when a specific system hierarchy is used, in this case the hierarchy of Figure 3.

The 'context' tags show in which Context the various stereotypes apply. Thus, we have the following:

- In the 'ACME Press Manufacturer' Context: 'Printing Press' is a System; 'Power Unit' is a Sub-system; 'Treadle' and 'Energy Storage' are both Assemblies; and 'Pedal', 'Connector Rod', Fly Wheel Axle' & 'Fly Wheel' are all Components.
- In the 'Press Power Systems Inc.' Context: 'Power Unit' is a System; 'Treadle' and 'Energy Storage' are both Sub-systems; and 'Pedal', 'Connector Rod', 'Fly Wheel Axle' & 'Fly Wheel' are all Assemblies.
- In the 'Wheel Castings Ltd' Context: 'Fly Wheel' is a System.

Elements are shown as belonging to different levels of the system hierarchy, depending on the Context considered. For each Context, the associated «context» tagged packages would contain model elements and diagrams that describe and/or reuse the holon at the appropriate level of granularity.

Extended applications and further work

The technique described here is not limited to hierarchical structures of holons or even to a single ontology. The authors focused on holons here because the length of the paper precludes extended examples and it is a familiar problem that provides a useful simplification for presenting the approach.

ISO 15288 [ISO 2015] recognises that "*there are an increasing number of systems that, from one or more aspects, are not hierarchical, such as networks and other distributed systems*." This implies the need for an approach that allows systems engineers to consider different parts of systems in different contexts at the same layer of granularity (which is not covered by Approaches 1 & 2 as described here).

Perry & Powley [2019] propose a technique using Domain and Context Ontologies to work in a single model with concepts that have (a) different names but the same meaning and/or (b) the same name but different meanings. The holon case discussed here applies Context Ontologies to (a). In further work, the intention is to combine both techniques to allow development of models that are both broadly and deeply integrated. This will address the non-hierarchical cases indicated by [ISO 2015].

Conclusion

This paper demonstrates that current MBSE techniques do not serve systems engineers well in terms of modelling holonic views of systems. This is a common problem, the resolution of which has significant benefits for both engineering practice and business outcomes. A simple solution is proposed that uses model elements with multiple stereotypes, each of which is tagged with a reference to the context in which that stereotype applies. Annotated diagram elements allow users of a diagram to see the context associated with the stereotypes.

The technique uses standard SysML or UML, both of which are widely used in Systems Engineering, and can be adapted for other languages and notations using descriptions to identify context. Although representations may differ between tools, the technique does not rely on specific tools capabilities; it is able to take advantage of additional tool features to automate usage, perform consistency checks and provide different visualisations.

The characteristics of this solution make it easy to adopt within an existing MBSE activity. Furthermore, the approach is entirely additive, meaning existing models can be extended whilst maintaining backwards compatibility. This ensures that stakeholders can continue to receive all the outputs from the model that they are accustomed to, without significant additional effort. Benefits will be seen through: greater model richness; and enhanced collaboration between roles through working on the same model at different abstraction levels. This may be particularly noticeable for systems engineers working in the same role, but in different contexts.

The authors have identified few downsides to the adoption of this technique. The only real issue encountered was that the tool used by the authors is unable to selectively display stereotypes, meaning that all of the multiple stereotypes must be shown whenever an element is used on a diagram. This leads to visual clutter that may be undesirable in some views. There is likely to be an adoption curve associated with integrating models and users may perceive an increase in complexity in the models. When a single model is extended to include holons, there will inevitably be an increase in the structural complexity of that model. However, this merely surfaces previously hidden real-world complexity that was obscured by having different models for different layers of abstraction. The extended model reveals knowable unknowns, which helps reduce residual uncertainty [HBR 1997].

MBSE practitioners, engineering teams, businesses and supply chains can all benefit from modelling holons. The technique described here is one method of achieving this and is distinguished by its simplicity and wide compatibility. A future technique that combines the proposal in this paper with that of [Perry & Powley 2019] will provide full coverage for all model integration scenarios, representing a significant step forward for MBSE. There may be some initial adoption challenges, but these will be offset by substantial medium to long-term benefits.

References

[Conway 1968] Conway, M. (1982). 'How Do Committees Invent?'. *In: Datamation magazine: April 1968.* F. D. Thompson Publications, Inc.

[Dijkstra 1982] Dijkstra, E. (1982). 'On the Role of Scientific Thought'. *In: Selected Writings on Computing: A Personal Perspective, pp. 60–66.* Berlin/Heidelberg, Germany: Springer-Verlag.

[Dori 2002] Dori, D. (2002). '*Object-Process Methodology: A Holistic Systems Paradigm*'. Berlin/Heidelberg, Germany: Springer-Verlag.

[HBR 1997] Courtney, H., Kirkland, J & Viguerie,P. (1997). 'Strategy Under Uncertainty'. *In: Harvard Business Review, HBR, Nov.-Dec. 1997 Issue.* [online] Available at: https://hbr.org/1997/11/strategy-under-uncertainty

[Holt & Perry 2018] Holt, J & Perry, S. (2018). *'SysML for Systems Engineering: A Model-Based Approach'*. 3rd ed. London: IET Publishing.

[Koestler 1967] Koestler, A. (1967). *'The Ghost in the Machine'*. London: Hutchinson & Co. Ltd.

[INCOSE 2015] International Council on Systems Engineering, INCOSE. (2015). '*Systems Engineering Handbook: A Guide for System Lifecyle Processes and Activities*'. Hoboken, New Jersey, USA: Wiley.

[ISO 2011] International Organization for Standardization, ISO (2011). '*ISO/IEC/IEEE 42010:2011 Systems and software engineering — Architecture description*'. Geneva, Switzerland: ISO.

[ISO 2015] International Organization for Standardization, ISO (2015). *'ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes'*. Geneva, Switzerland: ISO.

[ODUSD 2008] Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, ODUSD(A&T)SSE. *'Systems Engineering Guide for Systems of Systems'*. Washington, DC, USA: ODUSD(A&T)SSE.

[omg.org 2017a] omg.org. (2015). '*OMG System Modeling Language Specification Version 1.5*'. [online] Available at: https://www.omg.org/spec/SysML/1.5 [Accessed 03 May 2020].

[omg.org 2017b] omg.org. (2015). '*OMG Unified Modeling Language Specification Version 2.5.1*'. [online] Available at: https://www.omg.org/spec/UML [Accessed 03 May 2020].

[Perry & Powley 2019] Perry, S. & Powley, S. (2019). 'What's in a Name?' *In: INCOSE UK Annual Systems Engineering Conference (ASEC) 2019*. Ilminster, UK: INCOSE UK.