Adapting the Fact-Based Modeling Approach in Requirement Engineering

Yan Tang Demey

European Space Agency/ESTEC,
Keplerlaan 1, P.O. Box 299, NL-2200 AG Noordwijk, The Netherlands
yan.tang@esa.int

Abstract. Requirement Engineering plays a key role in developing complex space systems successfully. How to achieve efficient and effective information exchange during the whole life cycle of a space system is a challenge, which can be tackled by realizing *semantic interoperability* between involved partners (as presented in ECSS-E-TM-10-23A [1]). Using academic works related to fact based modelling (FBM), the European Space Agency (ESA) has initiated the development of a new knowledge management system (called Fact-based Modeling Unifying System or FAMOUS) to provide means to tackle semantic interoperability. The Agency has also organized an international working group of FBM experts to support this objective. In this paper, two Requirement Engineering use cases currently developed at ESA will be presented: one related to requirement analysis for spacecraft on-board software and the other one to requirement management for human spaceflight missions.

Keywords: Fact-based Modeling, Information modelling, Requirement Engineering, Semantic Interoperability and Ontology Engineering.

1 Introduction

The Fact-based Modeling (FBM, www.factbasedmodelling.org) approach, originated in Europe in the 1970s, is a "conceptual approach for modelling, transforming and querying information where all facts of interest are represented in terms of attribute-free structures known as fact types" [2].

FBM is a formal language based on First Order Logic. It combines g*raphical* notation, which makes it easy for modelers to express and visualize information and textual *verbalization* for depicting models into formal requirements that can be validated by the domain experts (not expert in conceptual modelling) prior to constitute the specification baseline of any customer/supplier contracts.

FBM is *conceptual*, addressing the semantics without precluding any implementation issues. Using FBM, modelers capture the needs from the stakeholders (i.e. the Customer) without constraining the resulting model to any technologies or solutions (e.g. UML or relational).

Though FBM marks a line between conceptual and implementation issues, the latter is not neglected either. FBM includes methods of transforming conceptual

R. Meersman et al. (Eds.): OTM 2014 Workshops, LNCS 8842, pp. 65–69, 2014.

models to lower-level structures for implementation. Suppose we want to have an information system containing a relational database management system (RDBMS) and exchange its content with others using XML in compliance with an interface control document (ICD) expressed as a XML schema (XSD), one can transform the FBM conceptual model into 1) a relational logical model and related SQL DDL for instantiation, and 2) a hierarchical logical model and related XSD, for exchanging.

Following the classical three-layer structure suggested in [3] (i.e. conceptual, logical and physical layers), FAMOUS provides the means to model in FBM and transform, in both directions, a conceptual model into the following two sets of models: 1) logical models (e.g relational, hierarchical, object oriented; 2) physical models (e.g. SQL and XMI for data repositories, but also XSD or man machine interface (MMI) specifications for users (humans or computers).

In addition, FBM also provides methods for transforming conceptual models to procedural programming code (e.g. in C++ and C#).

In the remaining of this paper, we will illustrate two use cases: one is about ontology-based requirement analysis for spacecraft on-board software in Sec. 2 and the other is requirement management for human spaceflight missions. Sec. 3. Conclusion and vision will be illustrated in Sec. 4.

2 Use Case 1: Ontology-Based Requirement Analysis for Spacecraft On-Board Software

Following the ECSS-E-ST-40C standards [4] [5], software resides at all levels of a space system, ranging from system functions to firmware. The standards can be viewed as a process model of software development in different project phases (i.e. 0, A, B, C, D, E and F). The customer derives requirements, which (in case they are not the requirements at the lowest level) break down at various levels in different phases. Every time when new requirements are derived, we shall ensure the *completeness* and *consistency* of the requirement and the requirement breakdown. Moreover, according to [6], we shall also ensure the *unambiguity*, *verifiability*, *modifiability* and *traceability* of requirements.

In order to achieve requirements of high quality, we first need to look into the source of a requirement, which is the set of semantic elements used by the functions of space system software. Since a decade ago, a reference on-board software architecture called OSRA has been discussed intensively in ESA's annual workshop on Avionics Data, Control and Software Systems (ADCSS, http://adcss.esa.int). This reference architecture is based on the definition of software architecture given by Bass et al. [7] – "The software architecture of a program or computing system is the structure or structures of the system, which comprises software components, the externally visible properties of those components, and the relationship among them…" Precisely speaking, OSRA is to realize the "visible properties" and the "relationship" in this definition.

OSRA consists of two sub-structures – *static* and *dynamic*. The static structure describes how functions, components (also called 'parts') and assemblies of software system are structured and dependent on each other. System hierarchy, interfaces and usage are modeled in the static structure. The dynamic structure involves active objects, such as tasks, processes and threads. A mapping between the static and dynamic structures needs be established.

Since FBM is capable of modeling both static and dynamic domain concepts, it provides an excellent tailoring facility of establishing this mapping. Taking into an account that FBM is conceptual, we can ensure the *unambiguity*, *consistency* and *completeness* of requirements realized by FBM's formalism methods, constraints checksum methods and derivation rules.

Another important characteristic provided by FBM to OSRA is the textual verbalization, with which we can map the requirements written in a natural language to formal expressions that can be defined using First-Order Logic or Description Logic, and vice versa. It ensures a requirement to be *verifiable*. The *Modifiability* and *traceability* are ascertained by the versioning methods in FBM.

Furthermore, the vision is to provide the engineers in charge of writing requirements with a toolset guiding them towards the elicitation of verified requirements, i.e. semantically correct with regard to the domain and knowledge of space systems.

3 Use Case 2: Requirement Management for Human Spaceflight Missions

A challenge in ESA's projects concerning human spaceflight missions beyond Low Earth Orbit (LEO) is to deal with nominal and *off-nominal* situations *autonomously*. The idea of involving ePartners (i.e. personalized crew support systems that are ubiquitous and maybe wearable) in astronaut-automation teams has been introduced and gradually getting mature in the campaigns of MARS-500[1], Mission Execution Crew Assistant project (MECA[2]), and now Human ePartner Agent Robot Teaming project (HEART).

In MECA-HEART, ePartners help a crew on a spaceflight mission with more effective collaborative operations by a social and cognitive solution based on a rich knowledge base. With the support of ePartners, the crew can understand contextualized constraints in a better way in order to cope with unexpected, complicated and potentially hazardous events.

The knowledge base of MECA-HEART contains the following components:

- Ontologies at different levels (i.e. an infrastructure ontology at a global level and domain ontologies) that are modeled in FBM. The ontology at the global level can contain, for instance, user and user-task context model. A domain ontology can be, e.g., a shared cognitive system infrastructure model or a requirements baseline model.

---

[1] http://www.esa.int/Our_Activities/Human_Spaceflight/Mars500

[2] http://www.esa.int/Our_Activities/Human_Spaceflight/Concordia/Experiments_2012-2013

- Data models, that are derived from the ontologies. For example, logical relational database models can be generated from the requirements baseline model.
- Datasets, which are the population of the data models, or, comply with the data models. Furthermore, the data from a dataset that complies with a data model can as well be annotated with domain ontologies.
- Process models (e.g. business process models, FBM derivation rules) that are properly annotated with the ontologies.

FBM is used to model the MECA-HEART ontologies and data models. Following the three-level FBM triangle architecture (i.e. meta-model level, model/schema level and instance/data level), instances in one triangle can be at the schema level of another. In MECA-HEART, this principle is applied to deal with complicated domains, such as requirements management.

The design specification of MECA-HEART requirements management consists of core functions and requirements that elaborate on these core functions [8]. Each design specification is derived, iteratively evaluated and refined accordingly. A cognitive system shall satisfy a set of requirements, each of which belongs to one or more *requirements baseline*s that are contextualized and organized by *use case*s. A requirement is justified by some *claim*s, which can be measured.

Together with this social, iterative and human-centric cognitive method of design specification, FBM provides methods of establishing relations between these key concepts, which can be further developed as linked models, each of which corresponds to a concept definition. Moreover, how to deal with semantics under the Open World Assumption in FBM tackles the challenge of coping with unexpected, complicated and potentially hazardous events.

4 Conclusion

In this paper, we have illustrated two use cases currently developed at ESA that are:

- used to demonstrate the power of FBM for capturing the stakeholder needs, transforming them into a formal specification that can be validated prior to contracting any suppliers;
- used to support the development of FAMOUS by acknowledging the ESA needs and ensuring that all required capabilities are implemented within FAMOUS.

Acknowledgement. Mr J.L. Terraillon is responsible for the OSRA development; Mr M. Wolff for the MECA-HEART; Mr S. Valera for FBM and FAMOUS. The work is supported by the internal research fellowship financed by the European Space Agency.

References

1. ECSS: Space System Data Repository ECSS-E-TM-10-23A. ESA/ECSS, Noordwijk (2011)
2. The FBM Working Group: Fact-Based Modelling Metamodel: Exchanging Fact Based Conceptual Data Models. Standardization Internal Report, European Space Agency, Nederlands (2014)
3. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
4. The S/W Working Group: Space Engineering - Software ECSS-E-ST-40C. ECSS, Noordwijk (2009)
5. The S/W Working Group: Space Product Assurance - software product assurance ECSS-Q-ST-80C. ECSS, Noordwijk (2009)
6. IEEE: IEEE Recommended Practice for Software Requirements IEEE Std 830-1993. IEEE (1993)
7. Bass, L., Clements, P., Rick, K.: Software Architecture in Practice, 3rd edn. Addison-Wesley (2012)
8. Westera, M., Boschloo, J., van Diggelen, J., Koelewijn, L., Neerincx, M., Smets, N.: Employing Use-Cases for Piecewise Evaluation of Requirements and Claims. In: Proceedings of the 28th Annual European Conference on Cognitive Ergonomics, New York, pp. 279–286 (2010)