Semantic Heterogeneity in the Formal Development of Complex Systems: An Introduction*

J. Paul Gibson[1] and Idir Ait-Sadoune[2]

[1] Département Logiciels-Réseaux, Telecom-SudParis, Évry, France
(SAMOVAR UMR 5157)
paul.gibson@telecom-sudparis.eu
[2] Supelec, Gif Sur Yvette, France
idir.aitsadoune@supelec.fr

System engineering is a complex discipline[1], which is becoming more and more complicated by the heterogeneity of the subsystem components[2] and of the models involved in their design. This complexity can be managed only through the use of formal methods[3]. However, in general the engineering of software in such systems leads to a need for a mix of modelling languages and semantics; and this often leads to unexpected and undesirable interactions between components at all levels of abstraction[4]. There are currently no generally applicable tools for dealing with this heterogeneity of interactions in the engineering of complex systems.

The heterogeneity exists in 3 different dimensions:

- Abstraction — as software engineers move from requirements to implementation, the semantics of the modelling languages move from the problem domain to the solution domain. Thus, it is quite common to see two or more languages used as the modelling moves from the abstract to the concrete (from the non-operational to the operational)[5].
- Systems of systems — software should not be isolated from the system and environment in which it is intended to operate. Systems are now engineered from components including software, hardware, wet-ware, etc . . . . The types of these subsystems can vary greatly: synchronous or asynchronous, deterministic or non-deterministic, parallel or sequential, etc . . . . It is unlikely that a single language is best suited to modelling such heterogeneity[6].
- Synthesis and analysis — the language in which one models a system is not usually the same language in which one reasons about the relationship between models, and the correctness of one model with respect to another[7].

There needs to be a clear separation of concerns, in these 3 dimensions, in order to facilitate re-usable models, methods, tools and software processes (methodologies). There also needs to be a simple way of integrating the different concerns.

* This work was supported by grant ANR-13-INSE-0001 (The IMPEX Project http://impex.gforge.inria.fr) from the Agence Nationale de la Recherche (ANR).

T. Margaria and B. Steffen (Eds.): ISoLA 2014, Part II, LNCS 8803, pp. 570–572, 2014.

As with object oriented architectures, low coupling and high cohesion are strong indicators of good design[8]. In a formal approach to system engineering we need low coupling between our different modelling languages and high cohesion within them. This can be best achieved by formal specification of good interfaces between the different types of semantics. Currently, the state of the art in heterogeneous system modelling is away from an ideal development environment, where the interfacing (between different semantic models) would be automated. This is the long-term objective, but we have only recently embarked on the journey towards this goal.

A previous thematic track initiated the research in the direction of the problems arising due to the heterogeneous nature of formal modelling[9] . Two of the published papers illustrated different techniques for managing the heterogeneity. In *Leveraging Formal Verification Tools for DSML Users: A Process Modeling Case Study*[10], we see a model driven development approach where formal methods are used to translate between different modelling languages. In *An Ontological Pivot Model to Interoperate Heterogeneous User Requirements*[11] we see a pivotal ontological model being used to manage heterogeneity of vocabularies and heterogeneity of formalisms during requirements modelling.

In this thematic track, we emphasis the complex nature of systems engineering and the need for automated tool support for integrating different semantic models. We note that the accepted papers discuss not only theoretical aspects, but also hint at methodological aspects which are key to industrial transfer of these approaches. In *Modelling and Verifying an Evolving Distributed Control System Using an Event-based Approach*[12] we see component-based system engineering where abstraction plays a key role in permitting the integration of different component types (specified using different semantic models), and reasoning about their dynamic interaction. In *Requirements driven Data Warehouse Design: We can go further*[13] we see that ontological reasoning mechanisms can used to automatically construct a set of requirements that are coherent and non-conflictual, even when expressed in a variety of modelling languages. Finally, the paper *On Implicit and Explicit Semantics: Integration issues in proof-based development of systems*[14] illustrates how formal ontologies can be used to model re-usable domain knowledge in an explicit manner, and how this knowledge can be used to prove the correctness of a system that operates within the domain environment. Refinement of the system model can then be used to guarantee correct construction of a system within the specified context.

The track demonstrates that progress is being made in this very challenging area. However, much more is left to do.

References

1. Stevens, R., Brook, P.: Systems engineering: coping with complexity. Pearson Education (1998)
2. Eker, J., Janneck, J.W., Lee, E.A., Liu, J., Liu, X., Ludvig, J., Neuendorffer, S., Sachs, S., Xiong, Y.: Taming heterogeneity-the ptolemy approach. Proceedings of the IEEE 91(1), 127–144 (2003)

3. Balarin, F., Lavagno, L., Passerone, C., Sangiovanni-Vincentelli, A., Sgroi, M., Watanabe, Y.: Modeling and designing heterogeneous systems. In: Cortadella, J., Yakovlev, A., Rozenberg, G. (eds.) Concurrency and Hardware Design. LNCS, vol. 2549, pp. 228–273. Springer, Heidelberg (2002)
4. Gibson, J., Mermet, B., Méry, D.: Feature interactions: A mixed semantic model approach. In: McGloughlin, H., O'Regan, G. (eds.) 1st Irish Workshop on Formal Methods (IWFM 1997). Electronic Workshops in Computing, Dublin, Ireland, BCS (1997)
5. Hazzan, O., Kramer, J.: The role of abstraction in software engineering. In: Companion of the 30th International Conference on Software Engineering, ICSE Companion 2008, pp. 1045–1046. ACM, New York (2008)
6. Baldwin, W.C., Sauser, B.: Modeling the characteristics of system of systems. In: IEEE International Conference on System of Systems Engineering, SoSE 2009, pp. 1–6. IEEE (2009)
7. Adrion, W.R., Branstad, M.A., Cherniavsky, J.C.: Validation, verification, and testing of computer software. ACM Computing Surveys (CSUR) 14(2), 159–192 (1982)
8. Briand, L.C., Wüst, J., Daly, J.W., Victor Porter, D.: Exploring the relationships between design measures and software quality in object-oriented systems. Journal of Systems and Software 51(3), 245–273 (2000)
9. Ait-Ameur, Y., Méry, D.: Handling heterogeneity in formal developments of hardware and software systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012, Part II. LNCS, vol. 7610, pp. 327–328. Springer, Heidelberg (2012)
10. Zalila, F., Crégut, X., Pantel, M.: Leveraging formal verification tools for dsml users: A process modeling case study. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012, Part II. LNCS, vol. 7610, pp. 329–343. Springer, Heidelberg (2012)
11. Boukhari, I., Bellatreche, L., Jean, S.: An ontological pivot model to interoperate heterogeneous user requirements. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012, Part II. LNCS, vol. 7610, pp. 344–358. Springer, Heidelberg (2012)
12. Attiogbé, C.: Modelling and verifying an evolving distributed control system using an event-based approach. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 575–589. Springer, Heidelberg (2014)
13. Khouri, S., Bellatreche, L., Jean, S., Ameur, Y.A.: Requirements driven data warehouse design: We can go further. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 590–605. Springer, Heidelberg (2014)
14. Ait-Ameur, Y., Gibson, J.P., Méry, D.: On implicit and explicit semantics: Integration issues in proof-based development of systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part II. LNCS, vol. 8803, pp. 606–620. Springer, Heidelberg (2014)