

A Semiotics Approach to Semantic Mismatches

Sabah Al-Fedaghi

Computer Engineering Department, Kuwait University, Kuwait
sabah.alfedaghi@ku.edu.kw

Abstract. This paper deals with modeling issues that are common to semiotics and software engineering. A major problem is that vague notions of modeling lead to difficulties in building real-world representations for use in the software development life cycle. Specifically, this paper focuses on building a consistent representation that eliminates semantic mismatches through analysis of semiotics. Semantic mismatches occur when the same term is associated with multiple concepts. Semiotics provides a good theoretical foundation for UML research, since a UML diagram can be considered a sign made up of signs. This paper introduces a new approach based on the so-called Flowthing Model (FM) to represent semiotics notions of sign, interpretant, and object, for use in studying the problem of semantic mismatches. The conclusion is that diagrammatic UML representation can lead to the appearance of such problems and that FM description provides separate streams such that no mixing can occur among terms as things that flow.

Keywords: Conceptual model, Semiotic triangle, Software engineering, Semantic mismatches, UML.

1 Introduction and Description of the Problem

A model-driven engineering approach [1] emphasizes the bridging of different technologies and the integration of various bodies of knowledge [2][3]. It is concerned with modeling standards, techniques and tools, organizational process changes, project estimation and cost [4], as well as interoperability. Modeling standards include systems (e.g., SysML), software (e.g., UML), and hardware.

A “model” of a “system”, in this context, is a construct that describes and explains the modeled domain as a phase in the development of a corresponding software system. This resultant description plays a crucial role as a blueprint from which phases of development evolve. The conceptual picture describes a real-world domain while excluding technical aspects and serves as a guide for the subsequent design phase. It provides a high-level representation of relevant entities and relationships among them in a system [6].

Building such a model is a major task involving knowledge management [7] to facilitate communication among stakeholders. “Model-driven development is a technology that aims to handle software development at a higher abstraction level using models as the main development artifact” [8].

This paper targets a set of challenges related to system development, including construction of a conceptual representation to serve as the foundation for software

development. It focuses on *analysis* of requirements in the development life cycle to specify the *what*: user requirements captured from the problem domain without technology-dependent details [9]. The design phase is concerned with specifying the *how*, when a software solution is analyzed and design artifacts are developed [10]. “‘Analysis’ designates some kind of understanding of a problem or situation, whilst ‘design’ is related to the creation of a solution for the analyzed problem” [11].

With a focus on the phase involving requirements analysis, this paper examines the specific subject of consistency through elimination of “semantic mismatches” in a system description. Without loss of generality, we describe a method that illustrates this problem and its proposed solution as used in the BRIDG (Biomedical Research Integrated Domain Group) project [12-13].

The BRIDG project is a health information system that uses UML as a modeling tool. The BRIDG project has brought together diverse standards communities “to clarify the semantics of clinical research across pharmaceutical, regulatory, and research organizations” [14]. The adopted model specifies declarative semantics through UML class diagrams that describe concepts and the relationships between them, while the business processes (the procedural semantics) are represented in UML activity and state diagrams.

Models in BRIDG were developed by teams constructing source models based on use-cases and existing standards to facilitate articulation of the semantics in UML representations. The outcome was a set of “harmonizable artifacts” for mapping relationship constructs, and use-cases that included both declarative (data) and procedural (activity and state diagram) representations of the semantics. All stakeholders then reviewed the concepts in the mapping to reach definitions of terms that satisfied all participants. “Thus, the harmonization process involved an iterative and cumulative process of knowledge assimilation and unification based on existing knowledge resources” [14].

This paper focuses on the attempt in BRIDG to build a consistent representation through the elimination of semantic mismatches by iteratively utilizing dual processes as follows:

- Analysis of “business processes” from the point in time a symptom appeared, along with an interpretation and subsequent analysis.
- Use of UML diagrams to maintain consistency between declarative and procedural semantics.

The paper proposes an alternative methodology to this oscillation of analysis of “business processes” by UML representation. It is based on the notion of streams of “flow things” (denoted by *flowthing*).

2 Semantics and the Semantic Ambiguity Problem

Consensus is a position of agreement reached by a team as a whole. It is often favored as a decision-making process by groups that support collaborative teamwork. Cooperative mechanisms in groups include sharing of information to facilitate a common “world map” view and objectives. In such a distributed information environment, it is necessary to coordinate cooperation, and that requires consensus among teams to synchronize information handling.

According to Fridsma et al. [14], consensus can be reached through *ambiguity*, *abstraction*, or *harmonization*. Consensus through ambiguity occurs when vague definitions are created that can be interpreted differently by different stakeholders, e.g., definitions purposely include vague statements on which all participants can agree. Consensus through abstraction is achieved through generalization of terms that can be interpreted or modified for the local needs of participants.

In the BRIDG project [14], harmonization was achieved as follows:

Experts examined the *business processes*... to determine causal and temporal relationships between the observed change in the patient’s clinical state and its association [with] the clinical trial activities. [...]

Second, ... the *activity diagrams* provide an orthogonal view of the declarative semantics of the class diagrams and provide a double check of the declarative representations in the model. [14]

In the context of this project, Fridsma et al. [14] introduced an underlying notion of “semantic mismatches” in terms of semiotics signs. Semiotics theory studies the meanings carried by signs (communication). A *sign* is “something standing for something else” [15], as shown in Fig. 1. The objects that can be signified by signs can be physical (a thing), conceptual (ideas), or even other signs.

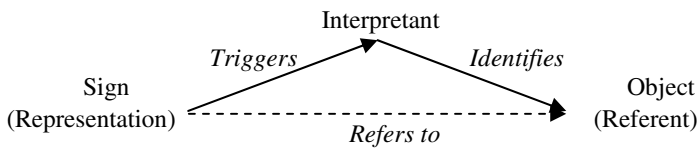


Fig. 1. Semiotic Triangle [15]

Semiotics has been used in many computer science studies. For example, according to Siau and Tian [17] “Semiotics, the study of signs, provides us good theoretical foundation for UML research as UML graphical notations are some kinds of signs.” A UML diagram can be considered a sign made of signs [18-19].

Using the Semiotic triangle, Fridsma et al. [14] examined semantic mismatches in which the same term is associated with multiple concepts. They resolved this problem by creating two new concepts, with new terms applied to each of the concepts. Fig. 2 shows an example of such semantic mismatch in terms of semiotic triangles.

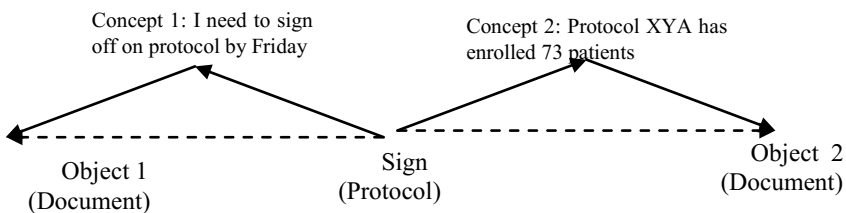


Fig. 2. Semantic mismatches: the same term associated with multiple concepts (from [14], with changes in terminology)

3 Flowthing Model

The Flowthing Model (FM) [20-22] represents some segment of reality as a web of interrelated *flows* that cross boundaries of intersecting and nested *spheres*. Ingredients in a flow include *flowthings* (things that flow), and flow systems (*flowsystems*). So-called objects, concepts, entities, and time are flowthings. A “thing” is defined as a flowthing: “what is created, released, transferred, arrived, accepted, and processed” while flowing within and among *spheres*. It has a permanent identity but impermanent form. A *flowsystem* constrains the trajectory of flow of flowthings. A particular flowsystem is the space/time for happenings and existence of flowthings. To flowthings, the flowsystem is formed from six discontinuities: being created, being released, being transferred, being arrived, being accepted, and being processed.

The ingestion of a change in flowthings happens in three ways:

1. Change in *sphere* (regions in the world) through being **released**, **transferred**, and **received** from one sphere to another.
2. Change in *existence* through being created (emerging) or de-created (extinguished). An a priori snapshot of the sphere of the **creation** does not contain the flowthing, and an a posteriori snapshot contains it.
3. Change in the *form* of one or more features through being **processed**, e.g., shape, color, size.

Flows connect six *states* (also called stages) that are exclusive for flowthings; i.e., a flowthing can be in one and only one of these six states at a time: transfer, process, creation, release, arrival, and acceptance, as shown in Fig. 4. We use *Receive* as a combined stage of *Arrive* and *Accept* whenever arriving flowthings are always accepted. A *state* here is a “transmigration field” of the flowthing that is processed, created, released, arrives, is transferred, and is accepted. In Fig. 3, we assume irreversibility of flow, e.g., released flowthings flow only to transfer.

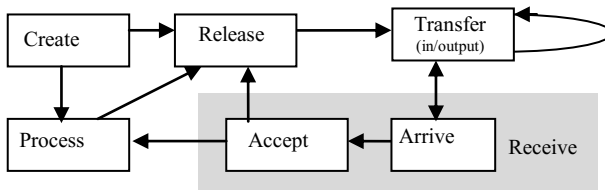


Fig. 3. Flowsystem

The exclusiveness of FM stages (i.e., a flowthing cannot be in two stages simultaneously) indicates synchronized change of the flowthing. A flowthing *cannot be changed in form and sphere simultaneously*. This is a basic systematic property of flowthings.

Initialization, stopping, and continuing of flows occur through *triggering*. Triggering is a control mechanism. It is the only linkage among elements in FM description besides flow and is indicated by dashed arrows. Synchronizations (e.g., join/fork) and logic notions (e.g., and/or) can be superimposed on the basic FM depiction.

4 Sign as a Flowthing

Sign, object, and interpretant in the semiotic triangle are viewed as flowthings in FM. Given a sign in a given sphere that includes the sign, its object, and interpretant, each has its own flowsystem, as shown in Fig. 4. Signs can be created, released, transferred, received, and processed, but for (material) objects, an interpretant can be only created and processed. For example, concepts cannot be transferred directly from one mind to another.

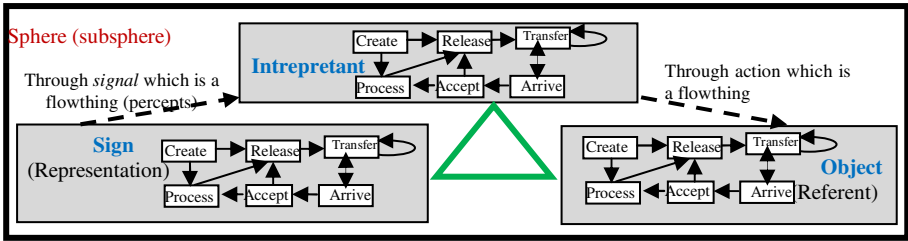


Fig. 4. The Semiotic triangle is formed from three spheres that include three different flowthings with their flowsystems

Assigning an object to a sign

Accordingly, the triangle can be supplemented with necessary flowsystems, as shown in Fig. 5. In the figure, a sign is created (appears/is generated; see circle 1). Assuming a physical sign, its appearance triggers (circle 2) physical signals, e.g., light (striking the retinas of the eyes), odor molecules, or pressure waves.

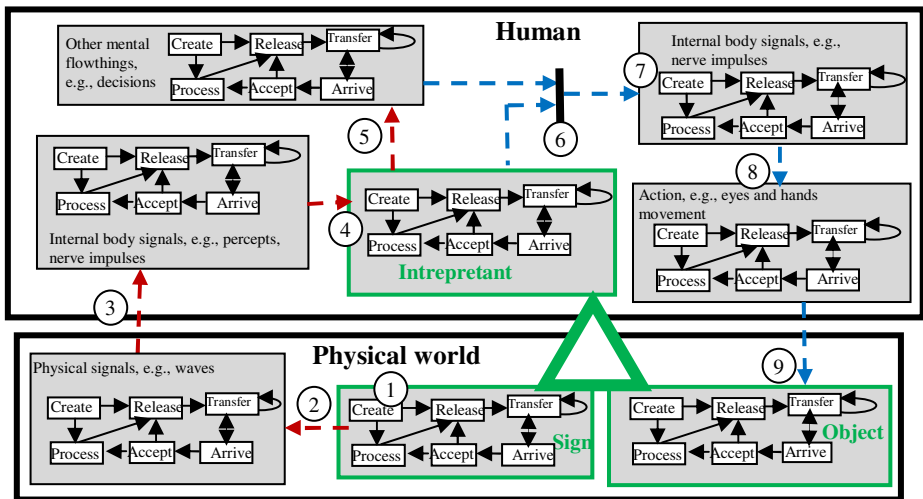


Fig. 5. The Semiotic triangle supplemented by the communication flowsystem between a human (body + mind) and the physical world.

These signals have their own flowsystems where they are created, received, etc., and they trigger (circle 3) the creation of a flowsystem of percepts/nerve impulses in the human body. These percepts/nerve impulses arrive at the brain, triggering creation of the interpretant (4), which in turn triggers other concepts (5).

With its “cluster” of associated concepts, the interpretant forms the “meaning” of the sign, thus directing attention to the thing to which the sign refers, assuming it is a physical object, through some type of action; formation of meaning triggers (6) creation of a flowsystem of percepts/nerve impulses in the human body. The vertical bar at (6) is borrowed from Petri nets notation to indicate a join of the effect of the interpretant and related concepts that form the “meaning” of the sign. Note that the purpose here is to demonstrate the methodology of the FM presentation and not to introduce a new cognitive theory of meaning. Any other description of the “mental” landscape can replace this part of the figure.

The “meaning” of the sign and intention to act on the object trigger internal body signals, e.g., nerve impulses (7), that in turn trigger body actions (8) that affect the object (9).

Assigning a sign to an object

On the other hand, assigning a sign to an object is modeled as shown in Fig. 6. Assuming physical object and sign, presence of the object (1) triggers the creation of physical signals, e.g., light (striking the retinas of the eyes), odor molecules, or pressure waves (2).

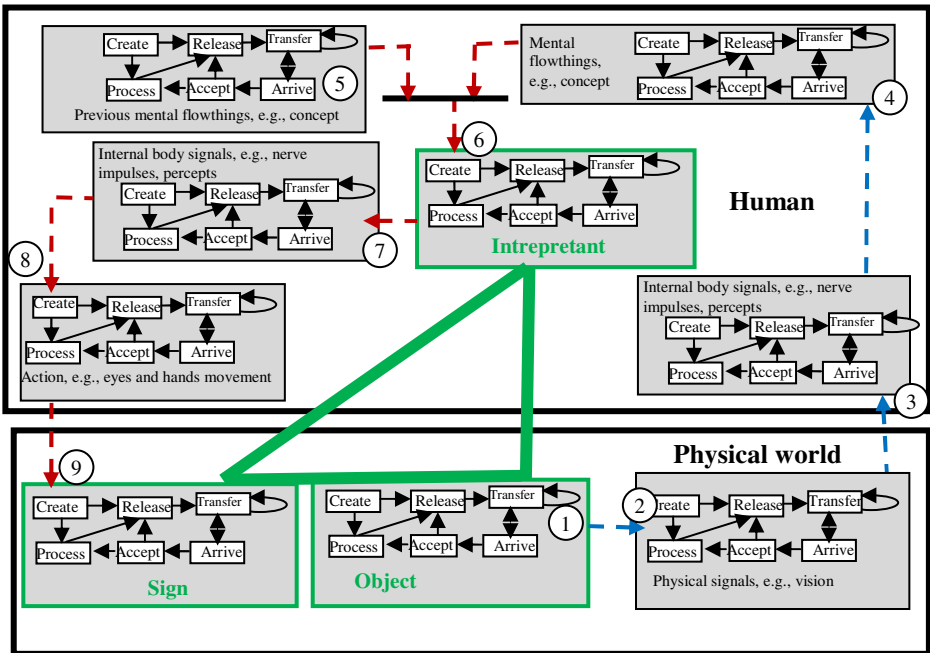


Fig. 6. The object causes creation of a sign

This, in turn, triggers the creation of a percepts/nerve impulses flowsystem in the human body (3), causing the generation of mental flowthings (4) triggered, originally, by the object, e.g., concepts of circles and lines form the concept of cylinder. Again, the purpose here is not to develop a particular cognitive process; rather, we describe assumed relationships among concepts to illustrate FM features and capability.

From the mental flowthings triggered by the signal coming from the object, and in the conjunction of mental flowthings already present in the mind from previous experiences (6), the interpretant is created. Notice that creation here does not necessarily mean no previous existence, but an emergence or appearance in the context of the current (sphere) experience caused by an encounter with the object.

The interpretant then triggers internal body signals (7) that trigger actions (8) that, in turn, trigger the creation of the physical sign, e.g., writing a word (9).

5 Consensus Re-visited

Now, in FM perspective, consider the notion advanced by Fridsma et al. [14] of semantic mismatches, in which the same term is associated with multiple concepts. Fridsma et al.’s description of the problem [14] implies that the two concepts are in two different minds (mentally separated spheres). It is difficult to imagine a person consciously using the same term for unrelated concepts. Similarly, in the context of a single organization, it is hard to imagine the same term being used for different concepts, but it might occur in disjointed or discrete suborganizational units that have indirect interaction and little personal interchange).

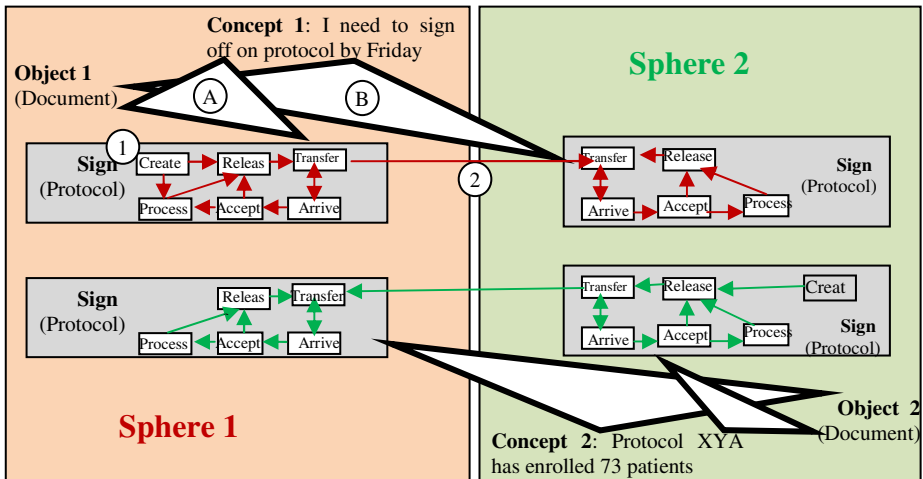


Fig. 7. The same term is associated with multiple concepts but each is recognized by its flow, under the assumption that the same sphere (a human) does not refer to multiple different concepts that identify multiple different objects by using one term

This means that a term corresponding to two concepts has been created in two different divisions of the organization. From the FM perspective, this means that the same term is used in two different spheres, as shown in Fig. 7, using the example given by Fridsma et al. [14] and shown previously in Fig. 2. Note that each sign is identified by its creator and its flow.

In the figure, the sign “protocol” is created by sphere 1 and flows to sphere 2. In sphere 2, this sign of “protocol” is received from sphere 1 and also another sign of “protocol” created internally in sphere 2. There is no confusion between the two signs because the FM flow description establishes the identity of each separately. Consider the sign “protocol” created in sphere 1 (circle 1). It is in the semiotic triangle labeled A. If it flows to sphere 2 (circle 2), it is in semiotic triangle B. It is just a sign in sphere 2, but its interpretant and project are still in sphere 1. Otherwise, by assumption, sphere 2 would not permit two terms to correspond to the same concept. Note that there is no *Create* stage in the upper flowssystem of sphere 2. As the sphere of a sign, sphere 2 can transfer, release, receive, and process this sign, but it is just a sign and has no corresponding concept in this sphere. If sphere 2 decides to attach a concept and object to this sign, then a mechanism is needed to import them from sphere 2. This is analogous to a postman who delivers a sealed *message* and simultaneously receives a personal message himself. Both are messages and may have the same form, but each has its own sender and receiver.

The problem in BRIDG that is discussed in this paper stems from the fragmentation of the UML representation that does not always tie each term to its stream of flow, like flows of electrical signals not bounded by wires. In FM, each kind of flowthing flows in its flowtssystem, and can be distinguished the same way electrical signals can be distinguished by following their flows from electrical sources. To illustrate such a feature of FM presentation, the next section contrasts the same example modeled in both UM and FM.

6 Contrasting the Two Representations

According to Fridsma et al. [14], the BRIDG model captures the procedural semantics of clinical research in activity diagrams that represent the processes in clinical trials. Fig. 8 shows an example of the procedural semantics represented in a UML activity diagram that functions like a storyboard.

The basic premise of the activity diagram is to present *events* with *multiple notations*, semantic *overloading*, and heterogeneous levels of descriptions. The point here is that this type of flowchart-like representation lacks an underlying web that preserves the tracking of things while they “transmigrate” among spheres, states, and existence/nonexistence. For example, a *request* that is sent vanishes into activities, databases, decisions, and processes, increasing chances of ambiguity such as occurs in the semantic mismatches discussed previously. This development of ambiguity contrasts with the FM representation with its systematic notions that all processes are categorized into stages and flowthings can be identified by flows.

Fig. 9 shows the FM representation corresponding to the activity diagram of Fig. 8 and drawn according to our understanding of the given description.

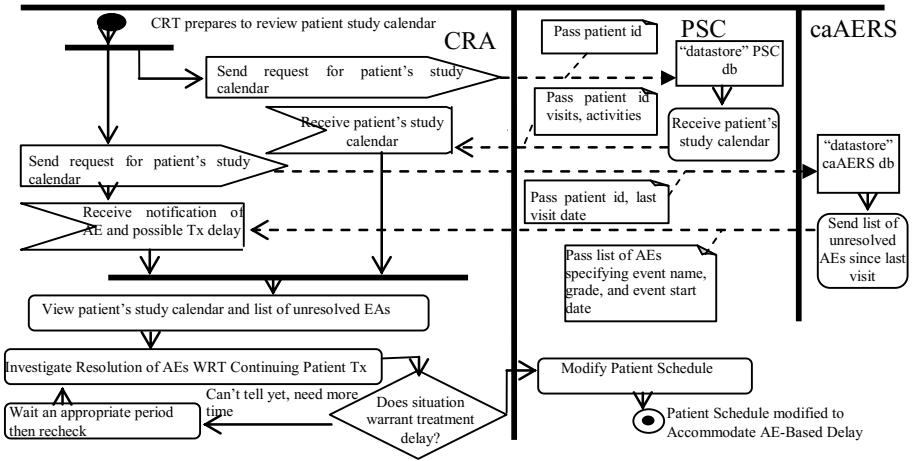


Fig. 8. Activity diagram of the example (from [14])

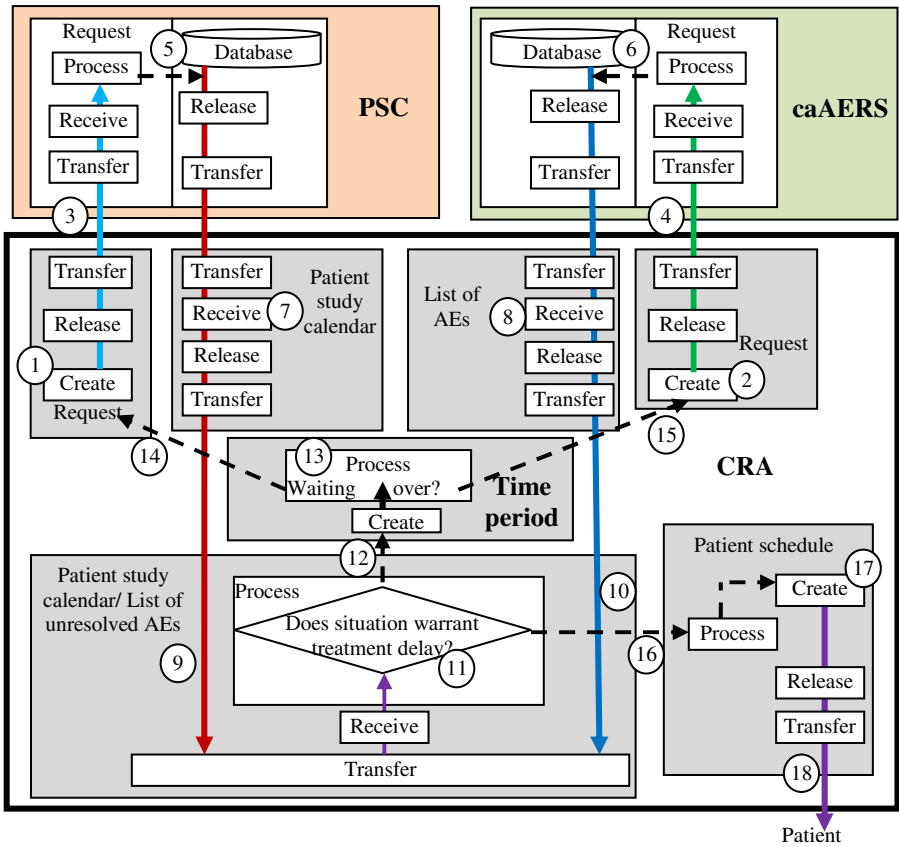


Fig. 9. FM representation that corresponds to Fig. 8

The CRT creates and sends two requests (circles 1 and 2 in the figure) to PSC and caAERS (circles 3 and 4, respectively). PSC and caAERS process these requests, triggering (5 and 6) the retrieval of “Patient study calendar“ and “List of AEs”, respectively. The subspheres “Patient study calendar“ and “List of AEs” in CRT receive these data and send them (circles 9 and 10) to the “Patient study calendar/ List of unresolved AEs” subsphere. Note that the flowthing in this subsphere is a *generalization* of the flowthings “Patient study calendar“ and “List of AEs”. Both of these last two flowthings are accepted as flowthings in “Patient study calendar“ and “List of AEs”. When they are processed (11), “Does situation warrant treatment delay?” can lead to “Can’t tell yet, need more time”, triggering (12) the creation of an appropriate time period that is processed (13). “Waiting over” triggers (14 and 15) the creation of new requests. Alternatively, “Situation does not warrant treatment delay” triggers (16) the processing of “Patient schedule” (available since it is received beforehand) to create a new one (17) that is sent to the patient. Note that in the online version of the paper, each flow is drawn in a different color.

Contrasting the resultant representations side by side points to the capability of the FM methodology to construct a consistent representation with its systematic notions where all processes are categorized into stages and flowthings can be identified by flows.

7 Conclusion

This paper deals with modeling issues that are common to semiotics and software engineering where vague notions of modeling lead to difficulties in building real-world representations. The paper introduces a flow-based model (FM) to represent semiotics notions of sign, interpretant, and object, for use in studying the problem of semantic mismatches. A study case of constructing source models based on UML use cases is re-casted in terms of the proposed methodology. The result points to the viability of FM to serve as a foundation for building a consistent representation.

References

1. Favres, J.M.: Towards a Basic Theory to Model Model-Driven Engineering. In: 3rd Workshop in Software Model Engineering (WiSME@UML), Lisbon, Portugal (October 2004)
2. Object Management Group: MDA Guide Version 1.0.1, <http://www.omg.org/>
3. Génova, G., Valiente, M.C., Marrero, M.: On the difference between analysis and design, and why it is relevant for the interpretation of models in Model Driven Engineering. *J. Object Tech.* 8(1), 107–127 (2009)
4. Blackburn, M.R.: What’s Model Driven Engineering(MDE) and how can it impact process, people, tools and productivity. Tech. Rep. Systems and Software Consortium, Inc. (2008)
5. Object Management Group: Unified Modeling Language Specification. Version 1.5 (March 2003), <http://www.omg.org/>

6. Autili, M., et al.: CHOReOS Perspective on the Future Internet and Initial Conceptual Model. Project report, version 1 (January 27, 2012), <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.220.3072&rep=rep1&type=pdf>
7. Brooks, F.P.: *The Mythical Man-Month*. Addison-Wesley (1975)
8. Silingas, D.: Model-Driven Service Choreographies for the Future Internet. In: *Future Internet Symposium 2012*, Vilnius, Lithuania, May 21-23 (2012)
9. Kaindl, H.: Difficulties in the Transition from OO Analysis to Design. *IEEE Soft.* 16(5), 94–102 (1999)
10. Høydalsvik, G.M., Sindre, G.: On the Purpose of Object-Oriented Analysis. In: *VIII Conference on Object-Oriented Prog. Syst. Lang. and Appl (OOPSLA 1993)*, September 26–October 1, vol. 28(10), pp. 240–255. ACM SIGPLAN Notices, Washington (1993)
11. Harel, D., Rumpe, B.: Meaningful Modeling: What’s the Semantics of ‘Semantics’? *IEEE Computer*, 64–72 (October 2004)
12. BRIDG (Biomedical Research Integrated Domain Group), <http://bridgmodel.nci.nih.gov/>
13. Hastak, S.: BRIDG Update, presentation (May 15, 2012), <http://bridgmodel.nci.nih.gov/>
14. Fridsma, D.B., Evans, J., Hastak, S., Mead, C.N.: The BRIDG Project: A Technical Report. *J. Am. Med. Inform. Assoc.* 15(2), 130–137 (2008)
15. Eco, U.: *A Theory of Semiotics*. Indiana University Press (1976)
16. Marcos, E., Marcos, A.: A Philosophical Approach to the Concept of Data Model: Is a DataModel, in Fact, a Model? *Inform. Syst. Frontiers* 3(2), 267–274 (2001)
17. Siau, K., Tian, Y.: A Semiotics View of Modeling Method Complexity: The Case of UML. In: *AMCIS 2005 Proceedings*. Paper 318 (2005), <http://aisel.aisnet.org/amcis2005/318>
18. Morand, B.: Modeling: Is it Turning Informal into Formal? In: Bézivin, J., Muller, P.-A. (eds.) *UML 1998*. LNCS, vol. 1618, pp. 37–48. Springer, Heidelberg (1999)
19. Génova, G., Valiente, M.C., Nubiola, J.: A Semiotic Approach to UML models. In: *CAiSE Workshops*, vol. 2, pp. 547–557 (2005)
20. Al-Fedaghi, S.: A Conceptual Foundation for Data Loss Prevention. *Int. J. Digital Content Tech. Appl.* 5(3), 293–303 (2011)
21. Al-Fedaghi, S.: Interpretation of Information Processing Regulations. *J. Softw. Eng. Appl.* 2(2), 67–76 (2009)
22. Al-Fedaghi, S.: Conceptualizing Effects, and Uses of Information. In: *The Information Seeking in Context conference (ISIC 2008)*, Vilnius. Lithuania, September 17-20 (2008)