Formal Framework for Ensuring Consistent System and Component Theories in the Design of Small Satellite Systems

Jules Chenou[1], William Edmonson[1], Albert Esterline[1], and Natasha Neogi[2]

[1] NC A & T State University
Greensboro, NC 27411
Email: {jchenou,wwedmons,esterlin}@ncat.edu
[2] National Institute of Aerospace
Hampton, VA 23666
Email: neogi@nianet.org

Abstract. We present a design framework for small-satellite systems that ensures that (1) each satellite has a consistent theory to infer new information from information it perceives and (2) the theory for the entire system is consistent so that a satellite can infer new information from information communicated to it. This research contributes to our Reliable and Formal Design (RFD) process, which strives for designs that are "correct by construction" by introducing formal methods early. Our framework uses Barwise's channel theory, founded on category theory, and allied work in situation semantics and situation theory. Each satellite has a "classification", which consists of tokens (e.g., observed situations) and types (e.g., situation features) and a binary relation classifying tokens with types. The core of a system of classifications is a category-theoretic construct that amalgamates the several classifications. We show how to derive the theory associated with a classification and the theory of the system core, and we show how to check whether a given requirement is derivable from or consistent with a theory.

1 Introduction

This research represents our ongoing effort to develop a model-based systems engineering methodology for small satellite systems that is reliable, formal and results in a "correct-by-construction" design. We present a knowledge-based design framework for ensuring that (1) each satellite has a consistent theory it can use to infer new information from information it perceives and (2) the theory for the entire system is consistent so that a satellite can infer new information from information communicated to it, and it can assess purported information from fellow satellites. The theories mentioned can be updated as insight is gained about the sensing satellites and the part of the world observed.

This research contributes to our development of a Reliable and Formal Design (RFD) process [1] [2], which strives for designs that are correct by construction by introducing formal methods early in the design process so that redesign may

be minimized. In previous work, we defined the structure for checking consistency and traceability of requirements in a formal manner [2]. In this paper, we expand on our definition of consistency in terms of category theory and information flow as it relates to designing systems of heterogeneous satellites that share information about the situations they observe.

Consider a particular small satellite S. A type (for S) is a feature of interest that S may observe in a monitored situation. We assume that we can identify a set of types that characterize S's sensory capabilities and, further, that we can form a table indicating the combinations of types that may occur together in an observed situation. In Section IV.A, we present an algorithm to derive a theory (for S) from this classification table. The algorithm can detect inconsistency. We may have to update S's classification table: we may find a new combination of types that may occur together (and so add a row to the table), or we may find that a previously postulated combination is in fact spurious (and so delete a row from the table). We may even find it necessary to introduce a new type for S. Whenever S's classification table is updated, we can run our algorithm and update S's theory (and check for consistency). S's theory allows it to infer additional information about a situation from information about that situation that it has perceived or received via communication with other satellites. S's theory also allows it to detect when purported information about a given situation from another satellite is incoherent from its (that is, S's) point of view or is inconsistent with the information S has about the situation in question.

We are concerned with a system of small satellites, so we consider the amalgamation of the classification tables of the several satellites in a given system as well as the theory inherent in this amalgamation. The amalgamation combines the content of the classification tables for the individual satellites and adds relations among types from several satellites. We describe this amalgamation in Section IV.C in terms of category theory. We can think of the system (with its classification table and theory) as the "whole" and the individual satellites as "parts". Again, in deriving the theory, we can determine whether it is consistent, and we can update the theory of the whole when the classification table for a part or a cross-part type dependency changes.

We describe, in an abstract manner, the algorithms and procedures that are used at a system-wide level in order to manipulate information and general knowledge for the satellite system. We do not suggest that the final implementation of the system of satellites will use these procedures. Rather, they are to be taken in the spirit of executable specifications. Later stages of the design process will refine the specifications into concrete designs that can be directly implemented.

The higher-level view taken here is based in the first instance on category theory, which captures abstract algebraic structures and their interactions in a coherent way (as categories) and also captures the relations between the categories. We make use of Barwise and Seligman's channel theory [3], which is an application of category theory to the "flow of information". Information is addressed here, not in terms of the amount of information as per the discipline

initiated by Shannon, but in the sense of the information content of an event or message. Flow here is not to be interpreted in a physical sense, but in the sense that X being α carries the information that Y is β.

Barwise and Seligman developed channel theory to explain the notion of a constraint, which was central in Barwise's account of "X being α carries the information that Y is β" in situation semantics. Barwise and Perry [4] developed situation semantics as a general theory of naturalized meaning and information. Devlin [5] provided a systematic presentation of situation semantics and the situation theory behind it. We exploit situation semantics and situation theory along with channel theory to give a rigorous and general account of the information satellites in a system of satellites have and share.

A *situation* is an ongoing happening in the world, while an *infon* is the basic item of information. An infon involves an n-place relation R, n objects appropriate for the corresponding argument places of R, a spatiotemporal location, and a polarity of 1 (indicating that the objects are thus related at the indicated place and time) or 0 (indicating otherwise). Alternatively, one may think of a situation as a partial function from tuples, with all the above components except polarity, to the codomain $\{0, 1\}$. A situation, unlike a possible world in the semantics of modal logics, targets only part of the world. A *real situation* is a part of reality (and considered a single entity) that supports an indefinite number of infons, while an *abstract situation* is a fixed set of infons.

Information flow is made possible by uniformities across relations between situations, that is, (as in channel theory) *constraints* (including natural laws and linguistic rules) that link various types of situations. Situation semantics addresses speech acts (utterances) and the "situatedness" of language use. It presents a theory of meaning that is relational in that language use relates situations: a linguistic unit (such as a declarative sentence) is uttered in an "utterance situation" whose descriptive content is some "described situation".

We retain the terminology of situation semantics for communication among small satellites since viewing satellite communication at a high level, in terms of utterance situations, allows us to abstract away unnecessary detail that adds nothing to the analysis. It also allows us to exploit notions relating to conversation that are not applicable at a lower level. A satellite broadcasting information delineates a spatiotemporal region in which there occurs an utterance situation (which also includes the satellites that are its physical neighbors given that an appropriate channel exists). We are interested in the information, its flow, conventions for taking turns, assumptions about content, and many other things at the level of speech and conversation. We are not interested in the physical aspects of communication and so eschew terms such as "broadcast".

The information of interest here concerns only the observed situations. The types relate only to observed situations, never to the satellites themselves. We do not address the issue of control actions taking by the satellites nor are we concerned with how information is extracted from signal. Real situations, with real objects and happenings, are observed although different satellites observe the same real situation from different perspectives and with different modalities.

We generally prefer "observe" for the verb relating to the source of information unless the emphasis is on the medium, in which case we tend to use "perceive".

We view the critical aspects of a small satellite as an intelligent knowledge based system (IKBS). An IKBS as proposed here has a syntactic aspect and a semantic aspect. The semantic aspect relates to the specific kind of information that the satellite can process and is represented by a finite number of "classifications." For generality, we allow for several classifications associated with the same satellite since it is common for a single satellite to have several sensing modalities or to use several fusion techniques. Each classification $\mathcal{A}$ (in the sense of channel theory) consists of a set $typ(A)$ of types and a set $tok(A)$ of tokens; a token $a \in tok(A)$ may be classified of type $\alpha \in typ(A)$ in classification A, written $a \vDash_A \alpha$ and called a *simple proposition*. So a classification $\mathcal{A}$ is a triple, $(tok(A), typ(A), \vDash_A)$. For a small satellite system, tokens are real situations (as observed by the satellites). All the classifications with which a given satellite is endowed are amalgamated into a core classification (as explained in Section IV) for that satellite, which we shall refer to as the classification of the IKBS.

The syntactic aspect of the IKBS of a satellite is a set of implication rules, which are the constraints mentioned above. Each rule is a sequent, of the form $\Gamma \vdash \Delta$, where Γ and Δ are sets of types. Suppose the classification in question here is $\mathcal{A}$. This sequent is *satisfied* by a token (real situation) a as long as, if $a \vDash_A \alpha$ for all $\alpha \in \Gamma$, then $a \vDash_A \beta$ for some $\beta \in \Delta$. A sequent is a constraint (for $\mathcal{A}$) if it is satisfied by all tokens in $tok(\mathcal{A})$. The deductive closure of the set of constraints is the IKBS's (or satellite's) theory (discussed above). The deductive closure of the set of rules is the IKBS's (or satellite's) theory (discussed above). Such a rule, if appropriately enabled, allows the satellite to infer new infons from its IKBS given other infons in the observed situation or the described situation associated with an utterance situation (where the utterance is by a neighboring satellite).

The next section describes our Reliable and Formal Design (RFD) process and how the techniques reported in this paper fit into this framework. Section III presents enough category theory and channel theory so that the reader may understand the rest of this paper. Section IV is the technical heart of this paper and presents techniques for amalgamating classifications, deriving a theory from a classification, and checking whether a requirement (encoded as a sequent) is derivable from a theory or may be consistently added to it. Section V considers how the satellites in a system maintain and communicate information on real situations. We assume that an IKBS may use its theory or the theory for the entire system to infer new information. Special attention is given to communication since an IKBS may fail to interpret an utterance or the content of what is uttered may be inconsistent with the information the IKBS has. Section VI concludes.

2 Reliable and Formal Design Process

The RFD (Reliable and Formal Design) process [2] follows a risk-tolerant philosophy that notionally will lead to a correct design with minimal-to-no redesign through the use of an agile and formal design process based on models. By integrating formal methods into the proposed design process at the appropriate levels, many design failures and integration challenges can be eliminated. Formal methods will provide automatic means for verification by translating requirements into a higher order logic language for which tools such as PVS [6] can perform consistency and traceability checks and proofs throughout the design process.

This RFD systems engineering process takes into account the fact that the design team is small and multi-disciplinary as well as the fact that system is complex and heterogeneous and is affected by its operational environment. Additionally, design of a complex system involves multiple disciplines that must interact symbiotically. This also implies that at the first step in the process each of the disciplines must have a global view of the system. As they proceed towards refinement, the design process becomes a local or discipline-specific activity, though always with a global perspective. The integration of formal methods into the design process of a complex system can reduce the need for a significant amount of revision during the system integration phase because of the "correct by construction" nature of the process. This leads to testing virtually at each level of refinement. Therefor, the theme of this design process from a systems engineering point of view is: *Think globally, design locally, and test virtually.*

At each level of abstraction, $\mathcal{A}_i$, the state of the RFD process can be represented by requirements, models, and simulations: $\mathcal{A}^i = (\mathcal{L}_n^i, \mathcal{L}_l^i, \mathcal{M}^i, \mathcal{S}_p^i, \mathcal{S}_b^i)$,where the information flow between these components is indicated with arrows as follows:

$$\begin{array}{ccccccc} \mathcal{L}_n^i & \Longleftrightarrow & \mathcal{L}_l^i & \Longleftrightarrow & \mathcal{M}^i & \Longrightarrow & \mathcal{S}_p^i \\ & & \Downarrow & & & & \\ & & \mathcal{S}_b^i & & & & \end{array} \tag{1}$$

Here

- $\mathcal{L}_n^i$ is the set of requirements written in natural language form
- $\mathcal{L}_l^i$ is the set of requirements written as a set of logical functions
- $\mathcal{M}^i$ is the system of interconnected models
- $\mathcal{S}_p^i$ is the set of simulations based on the parameters of $\mathcal{M}^i$.
- $\mathcal{S}_b^i$ is the set of simulations based on the logical description in $\mathcal{L}_l^i$.

The central result of this paper is the technique presented in Section IV.A for developing a theory for a satellite/IKBS from the classification table for it. As this addresses individual satellites, it is in the realm of local design, albeit at a very abstract level. The classification system used by an IKBS may be an amalgamation of several classifications. We form the classification for the entire

satellite system by a similar amalgamation (see Section IV.C) given the classifications of the component IKBS; the theory for the system is derived from this classification table. This is a return to a global perspective since the individual satellites with their sensing and communication capabilities were originally selected or built as required by the global mission. Theories for individual satellites and for the entire system are tested for consistency, which is virtual testing from the point of view of the system being developed.

The aspect in this knowledge-level design process emphasized in this paper is endowing each IKBS present within the distributed system with a theory in the form of a set of constraints. One determines for each IKBS the core classification with which it should be endowed and the classifications into which this core should be decomposed as "parts". Henceforth, we call the core classification for the IKBS simply the "IKBS classification". One then determines the types for each of the "part" classifications and the IKBS classification. These are the features of the IKBS's environment that it can sense or be informed about. One then produces a *classification table* for each "part" and for the core. The columns in the classification table are labeled with the classification's types, and we have a row for each abstract situation that (to our knowledge) may arise for that classification. A row has an '1' under a type if the abstract situation represented by that row supports that type; otherwise, it has a '0' under the type. Once the classification table for the IKBS classification is fixed, one can determine the IKBS's initial theory (a set of constraints) as described in Section IV.A. Determining the classifications for an IKBS can be done in a top-down fashion by first determining its core (or IKBS) classification and then determining the relative parts in a way that is sensitive to its physical attributes. Alternatively, this can be done in a bottom-up fashion, by mixing and matching basic classifications in an engineering repertoire and forming the IKBS classification once the parts are given.

Given a system of small satellites, we consider the amalgamation (the core) of the IKBS classifications. As with forming an IKBS classification, forming the core for the entire system is (in category-theory terms) a sum operation. Section IV.C shows how the classification table for the system core is constructed from the classification tables for the individual IKBSs. Again, we can form the initial theory of the system using the techniques described in Section IV.A. Combining theories raises the threat of inconsistency, and we check for inconsistency syntactically by determining whether, from the combined theory, we can derive some formula and its negation or, equivalently, we can derive the empty sequent $\langle \emptyset, \emptyset \rangle$. Consistency can also be defined semantically: see Section IV.B.

Designing a small satellite system is not just a matter of selecting the right kinds of satellites, developing their theories, and combining their reports at the knowledge level. One must also determine the overall situation, extended in both space and time, that the system will perceive. And one must formulate policies for how this overall situation will be divided into overlapping component situations allocated as regions of responsibility to the different satellites; these are policies for allocating sequences of overlapping situations to the various satellites. How

regions of responsibility should overlap depends both on what is monitored and the capabilities and theories with which the satellites are endowed. Significant overlap is expected when two satellites have complementary sensing modes or perspectives on a target.

How a satellite is actually deployed may give rise to additional constraints to be incorporated into its theory. And how several satellites are designed to collaborate may affect how the system core is formed and thus the theory for the system whole.

Once the satellite system is deployed, constraints will be added to the IKBSs, and their theories as well as the theory of the system as a whole will evolve. That is, new abstract situations will become apparent and could be added to the appropriate classification tables. One can check that the resulting theory remains consistent. One of the strengths of our approach is that, at any point in the lifetime of the systems, new requirements can be checked for consistency against current theories and, when consistent, used to augment those theories. At any time, the various theories must meet their requirements that have been collected over time. One way a theory can violate a requirement is by failing to include it. The solution in that case is to add the requirement to the theory and test for consistency. Another way a theory can violate a requirement is to be inconsistent with it, in which case the theory must be adjusted.

3 Category Theory and Channel Theory Overview

This section provides background for formal addressing the aggregation and flow of information in a small satellite system. The topics discussed here include category theory, channel theory, and the application of these theories to computational systems.

3.1 Category Theory

A category C consists of a class of objects and a class of morphisms (or arrows or maps) between the objects. Each morphism f has a unique source object a and target object b; we write$f : a \to b$. The composition of $f : a \to b$ and $g : b \to c$ is written as $g \circ f$ and is required to be associative: if in addition $h : c \to d$, then $h \circ (g \circ f) = (h \circ g) \circ f$. It is also required that, for every object x, there exists a morphism $1_x : x \to x$ (the identity morphism for x) such that, for every morphism $f : a \to b$, we have $1_b \circ f = f = f \circ 1_a$. It follows from these properties that there is exactly one identity morphism for every object.

A functor from one category to another is a structure-preserving mapping, preserving the identity and composition of morphisms. More exactly, if C and D are categories, then a functor F from C to D is a mapping that associates with each object $x \in C$ an object $F(x) \in D$ and, with each morphism $f : x \to y \in C$, a morphism $F(f) : F(x) \to F(y) \in D$. In addition, it requires that $F(id_x) = id_{Fx}$ for every object $x \in C$, and $F(g \circ f) = F(g) \circ F(f)$ for all morphisms $f : x \to y$ and $g : y \to z$. In category theory, a commutative diagram is a diagram of objects

(as vertices) and morphisms (arrows between objects) such that all directed paths in the diagram with the same start and end points lead to the same result by composition.

The classic presentation of category theory is in [6]. Two reasonably comprehensive and rigorous texts that are accessible to most readers with classical engineering mathematical backgrounds are [7] and [8]. A light introduction is provided by [9], while [10] and [11] are category-theory texts addressed specifically to computer scientists; [12] addresses category theory in the context of software engineering.

3.2 Channel Theory and the Flow of Information

Barwise and Seligman [3] presented a framework for the "flow of information" in (generally implicit) category-theoretic terms. They address the question, "How is it that information about any component of a system carries information about other components of the system?" For classifications $\mathcal{A}$ and $\mathcal{C}$, an infomorphism f from $\mathcal{A}$ to $\mathcal{C}$ is a pair of functions $(f^\wedge, f^\vee), f^\wedge : typ(\mathcal{A}) \rightarrow typ(\mathcal{C})$, and $f^\vee : tok(\mathcal{C}) \rightarrow tok(\mathcal{A})$, satisfying, for all tokens $c \in tok(\mathcal{C})$ and all types $\alpha \in typ(\mathcal{A})$, $f^\vee(c) \models_\mathcal{A} \alpha$ iff $c \models_\mathcal{C} f^\wedge(\alpha)$. In category theoretic terms, an informorphism is a kind of *Galois connection*. A *contravariant* Galois connection between $(tok(\mathcal{A}), typ(\mathcal{A}), \models_\mathcal{A})$ and $(tok(\mathcal{B}), typ(\mathcal{B}), \models_\mathcal{B})$ will be a pair (φ, ψ) of mappings $\varphi : tok(\mathcal{A}) \longrightarrow typ(\mathcal{B})$, $\psi : tok(\mathcal{B}) \longrightarrow typ(\mathcal{A})$ satisfying $a \models_\mathcal{A} \psi(b)$ if and only if $b \models_\mathcal{B} \varphi(a)$. An infomorphism between $(tok(\mathcal{A}), typ(\mathcal{A}), \models_\mathcal{A})$ and $(tok(\mathcal{B}), typ(\mathcal{B}), \models_\mathcal{B})$ however is a *covariant* Galois connection. It is a contravariant Galois connection between $(tok(\mathcal{A}), typ(\mathcal{A}), \models_\mathcal{A})$ and $(typ(\mathcal{B}), tok(\mathcal{B}), \models_\mathcal{B}^{-1})$ where $\models_\mathcal{B}^{-1} \subseteq typ(\mathcal{B}) \times tok(\mathcal{B})$ with $(\beta, b) \in \models_\mathcal{B}^{-1}$ if and only if $(b, \beta) \in \models_\mathcal{B}$.

Components of the system may be distant from one another in time and space, and the system can be made up of heterogeneous components. The system is "distributed" in this sense and not necessarily in the classical sense used in computer science. For example, the students, classrooms, scheduling system, and attendance records together form a distributed system related to students' attendance at a certain university.

An *information channel* is a family of infomorphisms with a common codomain, called the *core*. Essentially, a channel consists of a set $\mathcal{A}_1, ..., \mathcal{A}_n$ of classifications that represent the parts of the distributed system, a classification $\mathcal{C}$ (the core) that represents the system as a whole, and a set of infomorphisms $f_1, ..., f_n$ from each of the parts onto $\mathcal{C}$. Tokens in $\mathcal{C}$ are the *connections* of the system: a given token c in $\mathcal{C}$ connects the tokens it is related to by means of $f_1, ..., f_n$. Parts $\mathcal{A}_1, ..., \mathcal{A}_n$ carry information about each other as long as they all are parts of $\mathcal{C}$. Intuitively, an information channel is a part to whole $\mathcal{A}_i$-to-$\mathcal{C}$ informational relationship. Categorically, the core is a cocone in the category of classifications (objects are classifications and morphisms are informorphisms).

A *distributed system* D is a collection of elements that carry information about each other. Formally, D consists of an indexed class $cla(D)$ of classifications together with a class of infomorphisms, $inf(D)$, whose domains and

codomains are all in $cla(D)$. An information channel C covers distributed system D if and only if $cla(D)$ are the classifications of the channel and, for every infomorphism $f \in inf(D)$, there are infomorphisms from both the domain and codomain of f to the core of C such that these three infomorphisms are commutative in there interrelation. Basically, all classifications in D are "informational parts" of the core whose channel covers D.

Turning to regularities in a classification's types, let $\mathcal{A}$ be a classification and Γ and Δ be subsets of types in $\mathcal{A}$. Recall (from Section I) that a token a of $\mathcal{A}$ satisfies the *sequent* $\Gamma \vdash \Delta$ provided that, if a is a token of every type in Γ, then it is of some type in Δ. If every token of $\mathcal{A}$ satisfies $\Gamma \vdash \Delta$, then Γ is said to entail Δ and $\Gamma \vdash \Delta$ is called a *constraint* supported by $\mathcal{A}$. The set of all constraints supported by A is called the complete theory of $\mathcal{A}$, denoted by $Th(\mathcal{A})$. These constraints are systematic regularities, and it is by virtue of regularities among connections that information about certain component of a distributed system can be carried by other components into diverse parts of the system. Barwise and Seligman's summary statement of their analysis of information flow, restricted to the simple case of a system with two components, is as follows. "Suppose that the token a is of type α. Then a's being of type α carries the information that b is of type β, relative to channel C, if a and b are connected in C and if the translation β' of β entails the translation α' of α in the theory $Th(C)$, where $\mathcal{C}$ is the core of C" ([3], p.35).

3.3 Category Theory and Channel Theory for Computational Systems

Channel theory and category theory in general have had a significant impact in computer science, especially those aspects related to complex systems. Schorlemmer and Kalfoglou and their colleagues have applied channel theory in addressing semantic interoperability of federated databases [13] as well as the similar problem of ontology alignment [14] . Kent's Information Flow Framework (IFF) [15] also uses channel theory; IFF is being developed by the IEEE Standard Upper Ontology working group as a meta-level foundation for the development of upper ontologies. Spivak [16] presents a simple database definition language based on categories and functors and shows how to translate instances from one database schema to the other in canonical ways; Spivak and Kent [17] have also defined a category-theoretic model, OLOG, for knowledge representation. Finally, Diskin and Maibaum [18] support the claim that category theory provides a toolbox of design patterns and structural principles of real practical value for model driven software engineering. These research programs are generally at the knowledge level and as such fit well with the work reported here.

Goguen and Burstall's theory of institutions [19] is a categorical abstract model theory that formalizes the intuitive notion of a logical system, including syntax, semantics, and the satisfaction relation between them, which relates a semantic model to syntactically well-formed formulas that can be interpreted as true given the model. The meaning of the satisfaction condition of institutions is that truth is invariant under change of notation. Goguen has used institutions as

a basis for unifying and generalizing several approaches to information, including channel theory, Formal Concept Analysis, and Sowa's lattice of theories (ordered by inclusion on sets of derivable formulas) used for knowledge representation [20]. The work reported here gains impact by fitting into the general framework provided by the theory of institutions.

4 Amalgamating Classifications, Deriving Theories, and Checking Consistency

Figure 1 depicts the aspect of the RFD process presented here. On the left-hand side of the top, we have a specification in a natural language, $\mathcal{L}_n$, from which classification tables in the logical language $\mathcal{L}_l$ are encoded. From these tables, one generates a set of sequents (or constraints) using the algorithm presented in this section. The deductive closure of this set is the IKBS theory. Note that a theory is inconsistent if and only if one can derive the empty sequent, $\emptyset \vdash \emptyset$ from it. On the right-hand side of Figure 1, we have a requirement expressed in natural language $\mathcal{L}_n$. This is encoded as a sequent in $\mathcal{L}_l$. One then tests whether the sequent is entailed by the theory, is inconsistent with the theory (i.e., the theory with the sequent added is inconsistent), or neither. If it is entailed by the theory, we have increased confidence in both the theory and the requirement. If the requirement is inconsistent with the theory, then both the requirement and the theory could be suspect. If the requirement survives scrutiny, one redesigns the theory. Redesign in this approach focuses on the encoding and the correctness of the original $\mathcal{L}_n$ statements. If the sequent that encodes the requirement is not entailed by nor inconsistent with the theory, then one adds the sequent to the set of sequents whose deductive closure forms the theory.

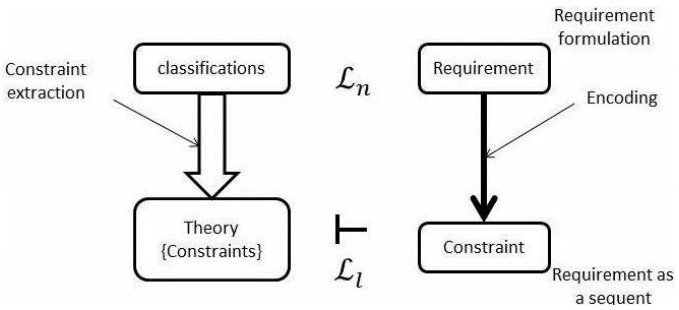

Fig. 1. RFD process as presented here.

If $\mathcal{A} = (tok(\mathcal{A}), typ(\mathcal{A}), \models_{\mathcal{A}})$ is a classification, we define

$$\alpha' = \{a \in tok(\mathcal{A}) | a \models_A \alpha\} \tag{2}$$

$$\Gamma^{\cap} = \Gamma' = \{x \in tok(\mathcal{A}) | \forall \alpha \in \Gamma, x \models_A \alpha\} = \bigcap_{\alpha \in \Gamma} \alpha' \tag{3}$$

$$\Gamma^{\cup} = \{x \in tok(\mathcal{A}) | \exists \alpha \in \Gamma, x \models_A \alpha\} = \bigcup_{\alpha \in \Gamma} \alpha' \tag{4}$$

It is straightforward that a token x satisfies the sequent $\langle \Gamma, \Delta \rangle$ if $x \in \Gamma^{\cap} \Rightarrow x \in \Delta^{\cup}$. Thus,$\langle \Gamma, \Delta \rangle$ is a constraint of $\mathcal{A}$ if and only if $\Gamma^{\cap} \subseteq \Delta^{\cup}$. An *implication* $\Gamma \rightarrow \Delta$ between types in $\mathcal{A}$ is a is sequent $\langle \Gamma, \Delta \rangle$ where Δ is non-empty. A subset $T \subseteq typ(\mathcal{A})$ respects an implication $\Gamma \rightarrow \Delta$ if $\Gamma \nsubseteq T$ or $\Delta \subseteq T$. An *implication* $\Gamma \rightarrow \Delta$ holds in $\mathcal{A}$ if and only if every subset of $typ(\mathcal{A})$ respects $\Gamma \rightarrow \Delta$ and so,if and only if $\Gamma^{\cap} \subseteq \Delta^{\cap}$

The Duquenne–Guigues [21] theorem provides a way to form a canonical basis of implications from sets of types that are pseudo-closed with respect to a closure operator of a classification $\mathcal{A}$. A set Γ of types is *pseudo-closed* with respect to a closure operator of a classification $\mathcal{A}$ if $\Gamma \neq (\Gamma')'$ and $(\Delta')' \subset \Gamma$ for every pseudo-closed $\Delta \subset \Gamma$. Note that in mathematics, a closure operator on a set S is a function $cl : P(S) \longrightarrow P(S)$ from the power set of S to itself which satisfies the following conditions for all sets $X, Y \subseteq S$, $X \subseteq cl(X)$, $X \subseteq Y \Rightarrow cl(X) \subseteq cl(Y)$ and $cl(cl(X)) = cl(X)$ As a base case for this definition, all minimal non-closed sets are pseudo-closed.

Algorithm 1 computes a basis of the theory of a given classification. The theory is the deductive closure of its basis. We use the Duquenne–Guigues theorem to compute part of the set of the constraints, namely, those that are implications. Note that in Algorithm 1 a sequent
$Gama \vdash \Delta$ where $\Delta = \emptyset$ is handled separately.

Algorithm 1 Constraint extraction procedure

1: **Input**: Classification $\mathcal{A}$
2: $Th(\mathcal{A}) = \varnothing$
3: For $\Gamma \subseteq typ(\mathcal{A})$
4: Compute $\Gamma^{\cup}$
5: If $\Gamma^{\cup} = tok(\mathcal{A})$ then $\Gamma \vdash \varnothing \in Th(\mathcal{A})$
6: If $\Gamma \vdash \Delta$ is in a Duquenne–Guigues basis of $\mathcal{A}$ then $\Gamma \vdash \Delta \in Th(\mathcal{A})$
7: End for
8: **Output** $Th(\mathcal{A})$

4.1 Logic of a classification

The formulas in the logic of a classification are sequents on the set of types. One can define a pseudo-negation and a pseudo-disjointion on the sequents.

A derivation or proof within this theory is as usual a sequence of sequents, where each of the sequents in the sequence is either a premise (element of the theory) or is deduced from previous sequents appearing in the sequence using one of Amstrong's rules [22], [23], described below:

Reflexivity If $\Gamma \subseteq \Delta$ then $\Gamma \vdash \Delta$

$$\frac{\Gamma \subseteq \Delta}{\Gamma \vdash \Delta}$$

Augmentation If $\Gamma \vdash \Delta$, then $\Gamma \cap \Gamma' \vdash \Delta \cup \Gamma'$

$$\frac{\Gamma \vdash \Delta}{\Gamma \cap \Gamma' \vdash \Delta \cup \Gamma'}$$

Transitivity If $\langle \Gamma, \Sigma \rangle$ and $\langle \Sigma, \Delta \rangle$ then $\Gamma \vdash \Delta$

$$\frac{\Gamma \vdash \Sigma, \Sigma \vdash \Delta}{\Gamma \vdash \Delta}$$

4.2 Consistency within a Classification Context

Consider the example classification given in Table below. We implemented Algorithm 1 in Python 2.7.5 and applied it to this classification, giving the basis of a theory as shown below.

$\vDash_A$	α	β	γ	σ
a_1	1	1	1	0
a_2	0	1	1	1
a_3	0	0	1	1

Th(A)	
$\{\alpha, \sigma\} \vdash \varnothing$	r_1
$\{\alpha, \gamma\} \vdash \{\beta\}$	r_2
$\varnothing \vdash \{\gamma\}$	r_3

Assume that a requirement of this system is expressed by the constraint $\{\alpha, \gamma\} \vdash \{\beta, \sigma\}$. We prove that this constraint is derivable from $Th(\mathcal{A})$.

1. $\{\alpha, \gamma\} \vdash \beta$ .....r_2, premise
2. $\beta \vdash \{\beta, \sigma\}\rangle$ ..... *Reflexivity*
3. $\{\alpha, \gamma\} \vdash \{\beta, \sigma\}\rangle$ .....1, 2, *Transitivity*

The proof is not always as short and easy as this. Generally, one would use an automatic or semi-automatic theorem prover such as PVS [24]. The derivability can also be shown semantically. Semantically, a sequent is valid in a classification $\mathcal{A}$ if each token of $\mathcal{A}$ satisfies the sequent. For a sequent $\Gamma \vdash \Delta$, this is equivalent to the inclusion $\Gamma^{\cap} \subseteq \Delta^{\cup}$. For example, for the sequent $\{\alpha, \gamma\} \vdash \{\beta, \sigma\}$, we have $\{\alpha, \gamma\}^{\cap} = \{a_1\} \subseteq \{a_1, a_2, a_3\} = \{\beta, \sigma\}^{\cup}$. In classical deductive logic, a consistent theory is one that does not contain a contradiction. The lack of contradiction can be defined in either semantic or syntactic terms. Here a model is taken to be a classification. The semantic definition states that a theory is consistent if and

only if it has a model, i.e., there exists an interpretation under which all formulas in the theory are true. This is the same sense used in traditional Aristotelian logic, although, in contemporary mathematical logic, the term "satisfiable" is used instead of "true". The syntactic definition states that a theory is consistent if and only if there is no formula p such that both p and its negation are provable from the axioms of the theory under the associated deductive system.

4.3 Sum of Classifications (Channel Core)

We introduced the notion of the core of an information channel in Section II.B, and we have viewed it as the whole that provides the amalgamation of the classifications of the parts. The core is formed as the category-theoretic *sum* of the classifications of the parts. Here we illustrate the sum of three classifications $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$, which is the core of the information channel containing the classifications. The sum $\mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ is the classification defined as follows:

1. The set $tok(\mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3)$ of tokens is the Cartesian product of $tok(\mathcal{A}_1)$, $tok(\mathcal{A}_2)$ and $tok(\mathcal{A}_3)$. Thus, the tokens of $\mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ are triples (a_1, a_2, a_3) of tokens, $a_1 \in tok(\mathcal{A}_1)$, $a_2 \in tok(\mathcal{A}_2)$ and $a_3 \in tok(\mathcal{A}_3)$.
2. The set $typ(\mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3)$ of types is the disjoint union of $typ(\mathcal{A}_1)$, $typ(\mathcal{A}_2)$ and $typ(\mathcal{A}_3)$. For concreteness, the types of $\mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ are pairs $\langle i, \alpha \rangle$, where $i = 1$ and $\alpha \in typ(\mathcal{A}_1)$ or $i = 2$ and $\alpha \in typ(\mathcal{A}_2)$ or $i = 3$ and $\alpha \in typ(\mathcal{A}_3)$.
3. The classification relation $\vDash_{\mathcal{A}_1+\mathcal{A}_2+\mathcal{A}_3}$ of $\mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ is defined by $(a_1, a_2, a_3) \vDash_{\mathcal{A}_1+\mathcal{A}_2+\mathcal{A}_3} \langle i, \alpha \rangle$ if and only if $a_i \vDash_{\mathcal{A}_i} \alpha, \forall i \in \{1, 2, 3\}$.

To make this example concrete, suppose that classifications $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ are given by the classification tables in following Table . Note that each abstract situation (row) is labeled. Then the first six rows classification table for $\mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3$ are shown below.

$\vDash_{A_1}$	α	β	δ	γ
a_1	1	0	1	0
a_2	0	1	1	0
a_3	0	1	0	0

$\vDash_{A_2}$	α	β	ρ
a_4	1	0	0
a_5	1	1	1
a_6	0	1	0

$\vDash_{A_3}$	α	ρ	σ	τ
a_7	0	1	0	1
a_8	0	1	1	0

$\vDash_{A_1+A_2+A_3}$	$\langle 1, \alpha \rangle$	$\langle 1, \beta \rangle$	$\langle 1, \delta \rangle$	$\langle 1, \gamma \rangle$	$\langle 2, \alpha \rangle$	$\langle 2, \beta \rangle$	$\langle 2, \rho \rangle$	$\langle 3, \alpha \rangle$	$\langle 3, \rho \rangle$	$\langle 3, \sigma \rangle$	$\langle 1, \tau \rangle$
(a_1, a_4, a_7)	1	0	1	0	1	0	0	0	1	0	1
(a_1, a_4, a_8)	1	0	1	0	1	0	0	0	1	1	0
(a_1, a_5, a_7)	1	0	1	0	1	1	1	0	1	0	1
(a_1, a_5, a_8)	1	0	1	0	1	1	1	0	1	1	0
(a_1, a_6, a_7)	1	0	1	0	0	1	0	0	1	0	1
(a_1, a_6, a_8)	1	0	1	0	0	1	0	0	1	1	0

4.4 Small Satellite Example

As an example, consider a system of three small satellites, S_1, S_2, and S_3, whose goal is to image the auroral ovals that exist around both magnetic poles of Earth and thereby study the impact of the solar wind on the magnetosphere. S_1 measures the intensity of the auroral brightness; if this is over a given threshold, then S1 measures the magnetic disturbance density and sends a message to S_2 and S_3 to begin imaging. S_2 and S_3 do the same thing from different perspectives, viz., take an image of the auroral oval to determine its extent.

For types for S_1, we distinguish auroral brightness below the threshold (LAB, "Low Auroral Brightness") and at or above the threshold (HAB, "High Auroral Brightness"). We arbitrarily partition the magnetic density measurement, a continuous magnitude, into a small number (viz., three) of ranges for simplicity, giving types LMD ("Low Magnetic Density"), MMD ("Medium"), and HMD ("high"). For S_2, we arbitrarily partition the auroral extent, a continuous magnitude, into three ranges: LAE_2 ("Low Auroral Extent"), MAE_2 ("Medium"), and HAE_2 ("High"). We recognize the same types for S_3 but use subscript "3" in place of "2": LAE_3, MAE_3, and HAE_3.

The classification table for S_1 is shown below. Some of the constraints derived from this table are the following, where (1) indicates that, if the auroral brightness is high, then there is a magnetic density measurement of high, medium, or low, and (2) indicates that the auroral brightness is (unconditionally) high or low.

1. $HAB \vdash LMD, MMD, HMD$
2. $\emptyset \vdash \{HAB, LAB\}$

The classification table for S_2 is shown below. The obvious constraints here are that there must be a measured auroral extent, low, medium, or high ($\emptyset \vdash \{LAE_2, MAE_2, HAE_2\}$), and it cannot be, for example, low and medium at the same time ($\{LAE_2, MAE_2\} \vdash \emptyset$). The classification table for S_3 is identical to that for S_2 but for the subscripts.

$SAT1$	HAB	LAB	LMD	MMD	HMD
1_a	0	1	0	0	0
1_b	1	0	1	0	0
1_c	1	0	0	1	0
1_d	1	0	0	0	1

$SAT2$	LAE_2	MAE_2	HAE_2
2_a	1	0	0
2_b	0	1	0
2_c	0	0	1

The classification table for the core will be analogous to the core classification produced in the previous example but will induce some constraints that cross system parts. At this point, it is convenient to enlarge our system to include a classification for the "part" that is observed by the satellites, namely, the magnetosphere and solar wind as well as the auroral oval produced by their interaction. The real situations classified with the types we have picked out so far certainly include these aspects of the environment and various types that characterize them. We consider here only four types that relate to this environmental part. Let AL indicate that the magnetosphere and solar wind are aligned (within some

level of tolerance), and let NAL indicate that they are not aligned. Let $COND$ indicate that this part is in a state where ionospheric conductivity could be exceptional, and let $NCOND$ indicate the denial of this. If we were to construct a classification table relating these types to types for the other three "parts" of the system (i.e., the satellite classifications), we could use our techniques to derive the following (and many more) constraints.

1. $COND \vdash AL$
2. $\{MMD, HAE_2, HAE_3\} \vdash AL$
3. $\{HMD, HAE_2, MAE_3\} \vdash COND$

Constraint (1) indicates that, if the situation is such that ionospheric conductivity could be exceptional, then the magnetosphere and solar wind are aligned. Constraint (2) indicates that, if the magnetic density is medium and the auroral extent is high from the perspective of both S_2 and S_3, then the magnetosphere and solar wind are aligned. And (3) indicates that, if the magnetic density is high and the auroral extent is high from the perspective of S_2 but medium from the perspective of S_3, then ionospheric conductivity could be exceptional.

5 Maintaining and Communicating Information on Real Situations

Consider now the *situation table* with the information an IKBS has on various real situations. We contrast the situation table with the classification table of a satellite, which is where design starts. The classification table has a column for each type relevant to the satellites sensing capability and the behavior of the monitored region. The rows indicate the combinations of types (columns) that may occur together in an observed situation. That is, the rows specify all the realizable abstract situations by indicating the types realized in them.

In contrast, the situation table has a row for each real situation observed. Like a classification table, a situation table has a column for each observable type, but the situation table needs information on more types. An IKBS has information on a situation not only by observation but also by virtue of utterance situations and by inferring information from information it already has. The descriptive content of an utterance may involve types observable by any satellite in the system, so the situation table must have columns for all these types. The IKBS has its own theory for inferring new information, but, since the IKBS is part of a distributed system, the theory of the whole, or core, is relevant as well for filling out entries in the situation table. Since the "whole" or core does not correspond to a physical entity over and above the "parts" (satellites), inferencing done with the theory of the core must be delegated to these parts. The easiest approach is to assume that all IKBSs have, besides their own theories, the theory of the core.

The general picture, then, has an IKBS, on observing a real situation, make an entry for it in its situation table, indicating what types were observed. When the descriptive content of an utterance relates to this situation, its table entry

may be filled by checking further types. It is also possible that observation may fill in an entry over an extended period. As information on the situation becomes available, various sequents of the local or core theory may allow further information to be inferred and recorded in the entry for the situation. A variation of this picture has the IKBS first finding out about the real situation in question via another's utterance. Note that issues we avoid in this paper include how real situations are denoted, how they are related as parts and wholes, and to what extent something true in a part is also true of the whole. These issues are addressed in the standard literature cited, and the part-whole relation has been extensively studied in terms of lattice structures [25].

The conditions for filling in an entry of the situation table because of a relevant utterance needs to be addressed in more detail since the IKBS may fail to interpret an utterance or the content of what is uttered may be inconsistent with the information the IKBS has. The descriptive content of what may be considered a normal utterance is what we call an *utterance proposition*, of the form $a : \Delta$, where a is a token and Δ is a set of types and type placeholders. For each type, we assume there is a corresponding placeholder $?_\alpha$. If $\alpha \in \Delta$, then part of what the utterance asserts is $a \models_{\mathcal{A}} \alpha$, while if neither $\alpha \in \Delta$ nor $?_\alpha \in \Delta$, part of what it asserts is a $a \not\models_{\mathcal{A}} \alpha$; we cannot have both $\alpha \in \Delta$ and $?_\alpha \in \Delta$. If $?_\alpha \in \Delta$, then the utterance says nothing about a being of type α. If the content of an utterance is consistent with an IKBS's theory (as discussed in Section IV.B), then it is included in the IKBS's situation table unless it is inconsistent with an entry already in that table. The situation table also includes results of observations, and it contains at most one entry for a given real situation (i.e., token). (As an IKBS's classification table constitutes in large part its semantic memory, its situation table constitutes its episodic memory.) We say that an utterance proposition $a : \Delta$ is in the situation table if the situation table has an entry for situation a and that entry records types and type placeholders as per Δ. If the situation table includes the utterance proposition $a : \Delta_1$ and the IKBS perceives an utterance whose descriptive content is $a : \Delta_2$ with the same token a, then the utterance proposition for a in the situation table can be updated with the information $a : \Delta_2$ as long as, for all types α of the IKBS classification, α is not in one of Δ_1 or Δ_2 but not the other. Violation of this condition indicates that the two utterance propositions are inconsistent. It may be the case, though, that α is in one but $?_\alpha$ is in the other, or α is not in one but $?_\alpha$ is in the other. Then the situation table entry for a, $a : \Delta_1$, is updated by replacing any $?_\alpha \in \Delta_1$ with α if $\alpha \in \Delta_2$ and by removing any $?_\alpha \in \Delta_1$ if $\alpha \notin \Delta_2$. If an IKBS succeeds in incorporating an utterance proposition into its situation table, we say that it has *interpreted* that utterance proposition.

In addition to the utterances just addressed, whose descriptive contents are utterance propositions, we allow utterances whose contents are *partial* utterance propositions; we call such utterances *p-utterances* and their descriptive content *p-propositions*. A p-proposition is again of the form $a : \Delta$, but now, for any type α, if $\alpha \notin \Delta$, then $a \not\models_{\mathcal{A}} \alpha$ is not being asserted; rather, the p-proposition has nothing to say about a being of type α. A p-utterance presents an occasion for

an IKBS to infer (using its theory) additional types for tokens, as illustrated by the paradigmatic case of inferring *fire* from *smoke*. A p-utterance could be used as a query, when one IKBS utters a p-utterance, and another utters a (normal) utterance in reply, providing missing types if they in fact are uniquely determined by the theory. A p-proposition $a : \Delta_1$ is inconsistent with an utterance proposition $a : \Delta_2$ in the situation table if there is at least one type α such that $\alpha \in \Delta_1$ but $\alpha \notin \Delta_2$. If the p-proposition is not inconsistent with any utterance proposition in the IKBS's situation table, then we say that the IKBS *interprets* that p-proposition; this is the case whether or not it is able to uniquely determine additional types for the token.

There are several ways an IKBS may fail to interpret the descriptive content of an utterance proposition or a p-proposition and hence fail to interpret the utterance or p-utterance itself. We say that an IKBS is *acquainted* with a token a if there is an utterance proposition $a : \Delta$ in its situation table, and we say that it is *acquainted* with a type α if α is among the types labeling the columns of its situation table. If an IKBS is unacquainted with a type, it is normally because that type is new and not yet incorporated into the IKBS's computational structures. An IKBS cannot interpret an utterance proposition or p-proposition $a : \Delta$ if there is a type $\alpha \in \Delta$ with which it is not acquainted, and it cannot interpret a p-proposition $a : \Delta$ if it is not acquainted with token a. Failure of interpretation due to lack of acquaintance amounts to a failure to understand. The other way interpretation can fail arises from inconsistency with content of the situation table, as described above.

6 Conclusion

As part of our engineering methodology for small satellite systems that is reliable, formal and results in a "correct-by-construction" design, we presented in this paper a knowledge-based design framework for ensuring that (1) each satellite has a consistent theory it can use to infer new information from information it perceives and (2) the theory for the entire system is consistent so that a satellite can infer new information from information communicated to it, and it can assess purported information from fellow satellites. The point of departure for our framework, from the previous RFD process, is Barwise's channel theory, founded on category theory, and allied work on situation semantics and situation theory. Each small satellite is viewed as an intelligent knowledge base system (IKBS) consisting of classifications in the sense of channel theory. A classification consists of tokens (e.g., observed situations) and types (e.g., features of a situation, such as a certain kind of event) as well as a binary relation that classifies tokens with types. Each IKBS has a semantic part, namely, the classifications with which it is endowed, and a syntactic part, which is a logic or theory of classification that allows each satellite to infer new information from observed and communicated situations. Communication (as per situation semantics) is viewed in terms of utterances; the descriptive content of an utterance is an assertion that a given token is of a given set of types. The core of a system of

classifications, as explained in this paper, is a category-theoretic construct that amalgamates the several classifications; the core and the individual classifications essentially form the "whole" and the "parts" of what is termed a channel. We show how to derive the theory for an IKBS and for the system core, and we show how to check whether a given requirement is derivable from or consistent with a theory.

References

1. Edmonson, W., Herencia-Zapana, H., Neogi, N., Moore, W., Ferguson, S.: Highly confident reduced life-cycle design process for small satellite systems: Methodology and theory. In: Complex Systems and Data Management Conference. (2012)
2. Edmonson, W., Chenou, J., Neogi, N., Herencia-Zapana, H.: Small satellite systems design methodology: A formal and agile design process. In: IEEE International Systems Conference. (2014)
3. Barwise, J., Seligman, J.: Information Flow The logic of Distributed Systems. Cambridge Tracks in Theoretical Computer Science. (1997)
4. Barwise, J., Perry, J.: Situations and Attitudes. MIT Press (1983)
5. Devlin, K.: Logic and Information. Cambridge University Press (1991)
6. MacLane, S.: Categories for the Working Mathematician. Springer-Verlag (1998)
7. Simmons, H.: An Introduction to Category Theory. Cambridge University Press (2011)
8. Awodey, S.: Category Theory. Oxford University Press (2010)
9. Lawvere, W.F., Schanuel, S.H.: Conceptual Mathematics A first Introduction to Categories. Bufflo Worshop Press. (1991)
10. Pierce, B.C.: Basic Category Theory for Computers Scientists. MIT Press. (1991)
11. Barr, M., Wells, C.: Category Theory for Computing Science. Prentice Hall (1998)
12. Fiadeiro, J.L.: Categories for Software Engineering. Springer (2005)
13. Schorlemmer, M., Kalfoglou, Y., Atencia, M.: A formal foundation for ontology-alignment interaction models. International Journal on Smantic Web and Information Systems (2007)
14. Kalfoglou, Y., Schorlemmer, M.: Formal support for representing and automating semantic interoperability. In: The Semantic Web, Springer, Heidelberg (2004)
15. Kent, R.E.: Semantic integration in the information flow framework. semantic interoperability and integration. In: Semantic Interoperability and Integration, Dagstuhl Seminar Proceedings, Wadern: Dagstuhl seminar Proceedings. (2005)
16. Spivak, D.I.: Functional data migration. Information and Computation (2012)
17. Spivak, D.I., Kent, R.E.: Ologs: A categorical framework for knowledge representation. PLoS ONE 7 (1) (2012)
18. Diskin, Z., Maibaum, T.: Category theory and model-driven engineering: From formal semantics to designs patterns and beyond. In: Workshop on Applied and Computational Category Theory
19. Goguen, J., Burstall, R.: Institutions: Abstract model theory for specification and programming. Journal of Association for Computing Machinery (1992)
20. Goguen, J.: Information integration in institutions. In: Moss, L. (Ed.) Memorial Volume for Jon Barwise. Indiana University Press, Bloomington. (2004)
21. Bazhanov, K., Obiedkov, S.: Optimizations in computing the duquenneguigues basis of implications. Annals of Mathematics and Artificial Intelligence (2014)
22. Armstrong, W.W.: Dependency structures of data base relationships. (1974)

23. Beeri, C., Dowd, M., Fagin, R., Statman, R.: On the structure of armstrong relations for functional dependencies. Journal of the ACM (1984)
24. Sam Owre, J.M.R., Shankar, N.: pvs: A prototype verification system. In: 11th International Conference on Automated Deduction (CADE). (1992)
25. Link, G.: Algebraic Semantics in Language and Philosophy. CSLI Lecture Notes No. 74 (1998)