

Towards a General Formal Framework of Coherence Management in RE

Alexander Borgida
Rutgers University
New Jersey, US
Email: borgida@cs.rutgers.edu

Ivan Jureta
University of Namur
Namur, Belgium
Email: ivan.jureta@unamur.be

Anna Zamansky
University of Haifa
Haifa, Israel
Email: annazam@is.haifa.ac.il

Abstract—Coherence Management refers to all efforts one needs to invest, in order to ensure that information shown in, and implied by a representation of requirements makes sense as a whole, is coherent. Coherence Management is an umbrella term we use to cover, and more importantly, stimulate research on relationships between identification, measurement, and action on phenomena which reflect tensions between information in requirements representations. Such tensions exist between information which is, for example, logically inconsistent, or stakeholders disagree on, or signals tradeoffs (meaning that improvement on some requirements, for instance, necessarily means some quantifiable (or not) deterioration of others). These tensions are an important topic of research in Requirements Engineering, and various methods have been proposed for the identification, measurement, and action on logical inconsistency in requirements models, on negotiating disagreements, and on settling tradeoffs. Despite focusing on related phenomena, these methods are different and each come with their own specific definition of when a representation of requirements is incoherent and what to do about it. This makes it hard to compare existing methods, design new ones, and choose those to apply when doing RE. In this short communication we outline our research agenda for developing a unified formal framework for the systematization and classification of Coherence Management efforts in the context of RE, as well as exploring their compatibility.

Index Terms—Requirements Engineering, Inconsistency Management.

I. INTRODUCTION

Incoherence of a requirements representation, of a model used in Requirements Engineering (RE), may occur as a result of opposing goals, lack of shared terminology between different stakeholders, or changes introduced during the evolution of the requirements [1], [2], [3]. While the notion of incoherence in RE is not yet fully understood, it is usually reduced to informal or formal inconsistency. Informal inconsistency involves having information which stakeholders and engineers find contradictory or involving tradeoffs, and which stimulates discussion and negotiation [4], [5], [6], [7]. Formal inconsistency has a narrow logical sense: descriptions such as requirements, goals, tasks, domain assumptions, etc. are represented as formulas in some logical language, and inconsistency is defined as the case when a contradiction can be derived from these formulas. Based on this notion, a wide variety of logic-based methods and tools have been proposed for identifying and managing formal inconsistency in models used during RE. Some prominent examples are Quasi-

classical Logic ([8], [9]), Techne ([10], [11]), CARL ([12], [13]), ViewPoints ([14], [15]) and many others ([16], [17], [18], [19], [20]).

It is important to recognise that both informal and formal inconsistency focus on the same range of phenomena in RE, namely tensions between information in requirements models, and about how to identify, measure, and act on these tensions. Despite focusing on related phenomena, proposed methods are different and each come with their own specific definition of when a representation of requirements is incoherent and what to do about it. This makes it hard to compare existing methods, design new ones, and choose the one to apply when doing RE. In this short communication we outline our research agenda for developing a unified formal framework for the systematization and classification of Coherence Management efforts in the context of RE, as well as exploring their compatibility.

II. MOTIVATION

Equating inconsistency in RE to a derivation (in classical logic) of a logical contradiction from a set of requirements, is too simplistic for two main reasons:

- *An RE model is more than just a logical theory.* Standard (and in particular classical) logic makes no distinction between different kinds of assumptions. Research in RE, on the other hand, has seen an explosion of ontological distinctions¹, which are helpful both in gathering requirements, and solving requirements problems. For instance, the influential Zave and Jackson's framework already distinguishes Requirements R, Specifications S and Domain Assumptions D, with the solution being to find S such that $D \cup S \vdash R$ under the condition that $D \cup S$ is consistent. Inconsistency in S and especially D seems, e.g., less acceptable than that in R, in early phases. Goal oriented approaches to requirements such as i-star further distinguish *goals*, *soft-goals* in R, and *functions* and *constraints* as operationalizations in S. These distinctions may further influence the way inconsistency should be handled.

¹Ontological distinctions are also used for inconsistency management in the database domain. When inconsistent databases are considered as FOL theories, the focus is on the inconsistency introduced by the atomic facts that make up the table contents, leaving out of scope the integrity constraints, which are assumed to be correct and consistent.

- *An RE inconsistency is more than just a logical contradiction.* Modern requirement modeling languages (RMLs) are designed to facilitate solving the RE problem by supporting (i) modelling, that is the representation of the problem and solution spaces; (ii) reasoning, that is the application of procedures to requirements models, to draw conclusions, for example, about requirements being satisfied, consistent, complete, etc; (iii) decision making, that is the recommendation of changes to the representation, based on the conclusions drawn. As such, they must be able to deal with variously (im)precise and (in)consistent expectations of the stakeholders, expressed in terms of beliefs, desires, and intentions, which may be contradictory but not in the strict classical logic sense. Moreover, RMLs are designed to handle non-functional requirements (NFRs), the conflicts between which (e.g., "high speed" vs "low memory use") cannot be expressed in terms of classical contradictions.

The first item above suggests that the categorisation of formulas as requirements, specifications, or otherwise, is important when managing coherence. The second item suggests that formal and informal inconsistency are related, and that understanding how they are related matters for coherence management.

III. ROADMAP

In this paper we propose a research roadmap for developing a general formal framework for dealing with tensions of information in requirements representations, in which a more fine-grained notion of tension and inconsistency can be captured.

Drawing inspiration from Thagard's coherence theory in cognitive science [21], we propose the term 'coherence' to refer to this fine-grained notion. Thagard's computational model of coherence describes the way in which we make sense out of conflicting pieces of information as an optimization problem. The basic elements in his model are notions such as concepts, propositions, parts of images, goals, actions, etc. Elements can cohere (or fit together) or incohere in various ways, imposing positive and negative constraints. The coherence problem then consists of dividing a set of elements into accepted and rejected sets in a way that satisfies most constraints, or *maximizes coherence*. Inconsistency in this model, therefore, is a particular kind of a negative constraint.

Our replacing the term 'consistency' with 'coherence' in the context of RE is therefore not accidental and bears important consequences. It signifies a shift from strict logical consistency towards the world of optimization, constraint satisfaction and utility theory, where desirability of goals and preferences of stakeholders are first-class citizens. Logical consistency is, therefore, a particular class of a more fine-grained notion of coherence.

Another problem we hope to address within our framework is a systematization of the jungle of inconsistency-tolerant logical formalisms. Despite relying on intuitively similar notions of inconsistency, current formalisms each come with their own specific definitions of when information in a requirements

model is inconsistent, and especially what conclusions one can draw from it. This makes it hard to compare existing methods and tools, design new ones, and choose those to apply when doing RE, leading to the need for a "jungle map": a list of desiderata of useful properties for coherence management formalisms in RE, against which all of the existing approaches, as well as new ones, could be classified, compared and analyzed.

As a first step, we can start from ([22], [23]), where a related problem was addressed in the context of (monotonic) paraconsistent logics, proposing a precise definition of "paraconsistency", as well as a desiderata list of formally defined criteria that a "good" paraconsistent logic should satisfy. Our aim for inconsistency-tolerant formalisms in RE is somewhat similar, with two crucial differences: the proposed properties should be (i) RE-specific (e.g., based on some influential RE ontology), and (ii) grounded in empirical/experimental verification. It is thus important to empirically pinpoint the needs of RE practitioners, mapping the various logical formalisms that are capable of (at least partially) satisfying them. In the long run, our results can also be a contribution to the challenging task of bridging formal and commonsense rational reasoning by combining logico-mathematical and empirical research.

Questions that we see as arising, and we plan to contribute on, include the following:

- *How exactly does or should the closure of an inconsistent set of requirements depend on the contents of that set?* There is no dependence, for example, if classical logic is used to represent requirements. More generally, any formalism which is explosive, where anything is the conclusion of an inconsistency, gives a useless answer to this question. The motive to answer this question is that it is important to avoid reaching conclusions which are entirely unrelated to what is already known, even if inconsistent.
- *Is it possible to compare formalisms in terms of which inconsistencies or tensions they can and cannot detect, in the same given requirements?* A formalism X may be able to detect inconsistency in a given set of requirements, yet another formalism Y may ignore it in that same set. And there may be inconsistency in that same set, which is ignored by X, but detected by Y. It is possible to define benchmarks, which can be used to compare formalisms in terms of their ability to detect inconsistency.
- *How can discussion and negotiation be reflected in a logical formalism?* Whether a statement can be concluded from an inconsistent set of requirements should depend on statements providing evidence to support that statement, and to defend it from counter-evidence. The question is how to relate dialectical reasoning and inconsistency, since argumentation, as a form of dialectical reasoning has been proposed as a general approach to negotiating requirements conflicts and validating requirements [7].
- *How are degrees of satisfaction of nonfunctional requirements related to truth values in logic?* Goal-oriented

RE notations such as i-star ([24]) allow requirements and specifications to be seen as *influencing* each other in a qualitative manner (e.g., "help" vs "hurt"). The only formalization of this we are aware of [25] is akin to a multi-valued logic based on subsets of 4 qualitative judgements {strongSupport, weakSupport, weakDenial, strongDenial}.

- *Non-functional requirements (qualities)* (NFR), are known to be extremely important in software development. They raise many issues:
 - *How to formalise tradeoffs between NFRs?* NFR are very frequently felt to be conflicting ("easy user access" vs "high security", "low latency" vs "high resilience"). As opposed to functional requirements, such conflicts are inherent in the nature of NFRs and may *never* get resolved. Decision making needs to be made based on choosing between alternative solutions [4]. In our opinion, the natural formalization of this would be expressed in terms of utility functions, optimization, constraint satisfaction, etc. Accordingly, these notions need to be integrated into standard logic-based approaches.
 - *How to account for gradeable adjectives in a logical formalisation of requirements?* The use of vague terms such as "high" (gradeable adjectives in Linguistics and Philosophy) is considered anathema in requirements, yet there has been work on their formal semantics, so that one could expect to discuss formally issues of (degrees of) conflict between them. Fuzzy logics (and their paraconsistent extensions) are the obvious candidates here, but linguists and philosophers have proposed different solutions, which should also be explored² [26], [27].
- *What other considerations does supporting reasoning with incoherent requirements induce?*
 - *Incompleteness:* Since one of the main motivations for supporting reasoning with incoherent requirements is the ability to hold discussions and make decisions at earlier phases, the framework should allow for incompleteness of the requirements ([28]).
 - *Explanation:* Note that although paraconsistent logics allow us to reason in the presence of inconsistencies, it is still important for requirements engineers to be aware whenever an inconsistencies/incoherences has arisen. In other areas involving model-building seen as formal theory development (e.g., ontologies), this has lead to tools that help authors "pinpoint" the minimal source of inconsistencies, and to offer explanations (as proofs) of how these axioms combine to lead inconsistency.
 - *Measuring incoherence:* As alluded to above for gradeable adjectives, there are interesting possibil-

²One of the important features of linguistics semantics is the strong influence of *context* on the meaning, so one might expect that formalization would also include notions of context.

ities in considering degrees of incoherence, for example, as a way of prioritizing conflict resolution. A good starting point is an adaptation of existing approaches of incoherence measurement to an inconsistency management framework ([29], [30], [31]).

IV. CONCLUSIONS

Consistency management, based on logical foundations, has long been an important concern of the RE community. In this paper we outline a roadmap for extending these logical foundations to support a more fine-grained notion of *Coherence Management*, referring to different types of tensions between information in requirements representations, including logical inconsistency, stakeholder disagreement, trade-offs, and many more. Our first aim is to develop a general framework in which the notion of coherence in the context of RE can be defined in precise terms, while being grounded in empirical/experimental verification. Our second aim is to propose a desiderata list of useful properties of coherence management formalisms, against which existing approaches can be compared, classified and systematized. It is our hope that the proposed research will initiate a discussion on providing logical foundations for notions from the world of optimization, constraint satisfaction and utility theory, which are becoming first-class citizens in modern RMLs.

REFERENCES

- [1] B. Nuseibeh, S. Easterbrook, and A. Russo, "Making inconsistency respectable in software development," *Journal of Systems and Software*, vol. 58, no. 2, pp. 171–180, 2001.
- [2] —, "Leveraging inconsistency in software development," *Computer*, vol. 33, no. 4, pp. 24–29, 2000.
- [3] A. Finkelstein, "A foolish consistency: Technical challenges in consistency management," in *Database and Expert Systems Applications*. Springer, 2000, pp. 1–5.
- [4] J. Mylopoulos, L. Chung, and B. Nixon, "Representing and using nonfunctional requirements: A process-oriented approach," *Software Engineering, IEEE Transactions on*, vol. 18, no. 6, pp. 483–497, 1992.
- [5] B. Boehm and H. In, "Identifying quality-requirement conflicts," *IEEE software*, vol. 13, no. 2, pp. 25–35, 1996.
- [6] B. Boehm, A. Egyed, J. Kwan, and R. Madachy, "Developing multimedia applications with the winwin spiral model," in *6th European Software Engineering Conference*. Springer, 1997, pp. 20–39.
- [7] I. J. Jureta, J. Mylopoulos, and S. Faulkner, "Analysis of multi-party agreement in requirements validation," in *17th IEEE International Requirements Engineering Conference*. IEEE, 2009, pp. 57–66.
- [8] P. Besnard and A. Hunter, "Quasi-classical logic: Non-trivializable classical reasoning from inconsistent information," in *Symbolic and quantitative approaches to reasoning and uncertainty*. Springer, 1995, pp. 44–51.
- [9] A. Hunter, "Reasoning with contradictory information using quasi-classical logic," *Journal of Logic and Computation*, vol. 10, no. 5, pp. 677–703, 2000.
- [10] I. Jureta, A. Borgida, N. A. Ernst, and J. Mylopoulos, "Techne: Towards a new generation of requirements modeling languages with goals, preferences, and inconsistency handling," in *RE*, 2010, pp. 115–124.
- [11] N. A. Ernst, A. Borgida, J. Mylopoulos, and I. J. Jureta, "Agile requirements evolution via paraconsistent reasoning," in *Advanced Information Systems Engineering*. Springer, 2012, pp. 382–397.
- [12] D. Zowghi, V. Gervasi, and A. McRae, "Using default reasoning to discover inconsistencies in natural language requirements," in *Eighth Asia-Pacific Software Engineering Conference*. IEEE, 2001, pp. 133–140.
- [13] V. Gervasi and D. Zowghi, "Reasoning about inconsistencies in natural language requirements," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 14, no. 3, pp. 277–330, 2005.

- [14] S. Easterbrook and B. Nuseibeh, "Using viewpoints for inconsistency management," *Software Engineering Journal*, vol. 11, no. 1, pp. 31–43, 1996.
- [15] S. Easterbrook and M. Chechik, "A framework for multi-valued reasoning over inconsistent viewpoints," in *Proceedings of the 23rd international conference on software engineering*. IEEE Computer Society, 2001, pp. 411–420.
- [16] O. Rodrigues, A. S. d'Avila Garcez, and A. Russo, "Reasoning about requirements evolution using clustered belief revision," in *Advances in Artificial Intelligence, 17th Brazilian Symposium on Artificial Intelligence*, 2004, pp. 41–51.
- [17] G. Spanoudakis and A. Zisman, "Inconsistency management in software engineering: Survey and open research issues," *Handbook of software engineering and knowledge engineering*, vol. 1, pp. 329–380, 2001.
- [18] A. Russo, R. Miller, B. Nuseibeh, and J. Kramer, "An abductive approach for analysing event-based requirements specifications," in *Proceedings of the 18th International Conference in Logic Programming ICLP*, 2002, pp. 22–37.
- [19] E. Bagheri and A. A. Ghorbani, "Experiences on the belief-theoretic integration of para-consistent conceptual models," in *19th Australian Conference on Software Engineering*. IEEE, 2008, pp. 357–366.
- [20] K. Mu, W. Liu, Z. Jin, R. Lu, A. Yue, and D. Bell, "Handling inconsistency in distributed software requirements specifications based on prioritized merging," *Fundamenta Informaticae*, vol. 91, no. 3, pp. 631–670, 2009.
- [21] P. Thagard and K. Verbeurgt, "Coherence as constraint satisfaction," *Cognitive Science*, vol. 22, no. 1, pp. 1–24, 1998.
- [22] O. Arieli, A. Avron, and A. Zamansky, "What is an ideal logic for reasoning with inconsistency?"
- [23] ———, "Ideal paraconsistent logics," *Studia Logica*, vol. 99, no. 1–3, pp. 31–60, 2011.
- [24] E. Yu, "Modelling strategic relationships for process reengineering," *Social Modeling for Requirements Engineering*, vol. 11, 2011.
- [25] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Formal Reasoning Techniques for Goal Models," *J. of Data Semantics*, vol. 1, pp. 1–20, 2004.
- [26] C. Kennedy, "Vagueness and grammar: The semantics of relative and absolute gradable adjectives," *Linguistics and philosophy*, vol. 30, no. 1, pp. 1–45, 2007.
- [27] I. Douven, L. Decock, R. Dietz, and P. Égré, "Vagueness: A conceptual spaces approach," *Journal of Philosophical Logic*, vol. 42, no. 1, pp. 137–160, 2013.
- [28] J. Horkoff, R. Salay, M. Chechik, and A. Di Sandro, "Supporting early decision-making in the presence of uncertainty," in *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*. IEEE, 2014, pp. 33–42.
- [29] B. Barragáns-Martínez, J. Pazos-Arias, and A. Fernández-Vilas, "On measuring levels of inconsistency in multi-perspective requirements specifications," in *Proceedings of the 1st Conference on the Principles of Software Engineering*, 2004, pp. 21–30.
- [30] A. Hunter and S. Konieczny, "Approaches to measuring inconsistent information," in *Inconsistency tolerance*. Springer, 2005, pp. 191–236.
- [31] J. Grant and A. Hunter, "Measuring inconsistency in knowledgebases," *Journal of Intelligent Information Systems*, vol. 27, no. 2, pp. 159–184, 2006.