Usefulness of Inconsistency in Collaborative Knowledge Authoring in Semantic Wiki

Weronika T. Adrian, Grzegorz J. Nalepa and Antoni Ligęza

Abstract Inconsistency in knowledge bases traditionally was considered undesired. Systematic eradication of it served to ensure high quality of a system. However, in case of semantic wikis, where distributed, hybrid knowledge bases are developed and maintained collectively, and inconsistency appears to be an intrinsic phenomena. In this paper, we analyze inconsistency in a semantic wiki system in terms of its origin, level, type, and significance. We claim that in some cases inconsistency should be tolerated and show examples where it can be used in a constructive way.

Keywords Semantic wikis · Knowledge engineering · Inconsistency

1 Introduction

Traditional approach to inconsistency in knowledge-based systems considered it an anomaly [25]. One of the main reasons for that is the "principle of explosion" (ECQ, from Latin: *Ex contradictione sequitur quodlibet* which means "from a contradiction, anything follows"). If anything can be entailed from a set of inconsistent statements, then the inconsistent knowledge base becomes unusable. Therefore, numerous methods and techniques have been developed to suppress inconsistency, either by rejecting contradictions (removing, forgetting, etc.) or by searching for a consensus to restore consistency.

With the advent of modern Web-based technologies, there is a growing intensity of collaboration on the Web. Wikipedia-like portals, recommendation systems, or community websites are examples of modern knowledge bases. New challenges are

W.T. Adrian (✉) · G.J. Nalepa · A. Ligęza
AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: wta@agh.edu.pl

G.J. Nalepa
e-mail: gjn@agh.edu.pl

A. Ligęza
e-mail: ligeza@agh.edu.pl

A.M.J. Skulimowski and J. Kacprzyk (eds.), *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*, Advances in Intelligent Systems and Computing 364, DOI 10.1007/978-3-319-19090-7_2

posed by distributed knowledge authoring, increased use of mobile devices, dynamic changes of the knowledge, and use of hybrid knowledge representation with mixed levels of formality [6]. Quality of knowledge is often evaluated collectively, by discussion, negotiations, and voting.

It is impractical to treat such knowledge bases same as centralized homogeneous systems, where consistency was an important quality factor. We claim that in semantic wiki systems, struggling for consistency can be ineffective, and can suppress such desirable phenomena as collaborative synergy and fast development of knowledge. Thus, inconsistency should be accepted and incorporated into reasoning rather than removed or ignored. This paper is an enhanced version of the paper presented at the KICSS2013 conference [3].

The paper is organized as follows: In Sect. 2, we introduce the motivation for our work. Then we review selected approaches to handle inconsistency, taking into consideration various aspects and levels of it in Sect. 3. A conceptualization of our semantic wiki environment is proposed in Sect. 4, followed by an analysis of inconsistency in it in terms of origin, types, and significance. This constitutes a starting point for a discussion on tolerating inconsistency in Sect. 5 and presentation of an exemplary use case in Sect. 6. Conclusion and future work is outlined in Sect. 7.

2 Motivation

Verification and validation of knowledge-based systems [13] is a mature field in which numerous solutions, both classic and recent [28], have been proposed. Verification challenges and algorithms depend on selected knowledge representation and reasoning within the considered system. XTT2 [36] is a logic-based representation for rule-based systems, which allows to visually model, refine, and execute modularized rule bases. Moreover, a formal analysis and verification is possible [32]. Such a representation could be adapted for modeling distributed knowledge bases (e.g., with use of semantic wikis [1, 31]).

Specific issues related to collective knowledge engineering [34] as well as verification [5] in such environments have been investigated. It appears that to some extend it would be beneficial to use some of the existing verification solutions. However, inconsistency in collaborative settings appears to be intrinsic phenomena, so the verification methods should be adapted accordingly. When inconsistency arises, the system should automatically and appropriately react, i.e., recognize if it is undesirable, or if it may be accepted and used constructively. Incorporating inconsistency into reasoning and taking advantage of it is not a new idea. In [15], several practical use cases were given and a general framework for including inconsistency was presented, and in [39], usefulness of inconsistency in software development has been discussed.

In order to develop reasonable methods of verification and inconsistency handling for knowledge engineering in semantic wikis, we aim to analyze various aspects, levels, and types of inconsistency. In the following section, we ground our discussion in overview of existing approaches to inconsistency in general, and then proceed to inconsistency analysis in semantic wiki system.

3 Preliminaries

3.1 Basic Concepts

One can find several interpretations of inconsistency that are reflected in various definitions. Intuitively, inconsistency appears when a set of sentences (formulas, theorems, beliefs) cannot be true at the same time. More precisely,

A formal system is inconsistent if there is a formula ϕ such that both ϕ and $\neg\phi$ are theorems [17]. Alternatively,

Axiomatic system is inconsistent if for a given set of axioms Γ (relative to a given logical language $\mathcal{L}$) a formula ϕ can be entailed ($\Gamma \models \phi$) and similarly $\neg\phi$ can be entailed ($\Gamma \models \neg\phi$) [23]. Finally,

A contradiction between two statements is a strong kind of inconsistency between them, such that one must be true and the other must be false.[1]

If model-theoretic semantics is concerned, then the knowledge base or its corresponding theory is inconsistent if it does not have a model [12]. There are also less formal terms in use, not necessarily equivalent to logical inconsistency, for instance incoherent data or incompatible conceptualizations [23].

3.2 Formal Representation of Inconsistency

In order to formalize inconsistency handling, there must be a formal representation of inconsistency itself. In [11], where logics are defined as "formal systems consisting of a language L (in the form of a set of formulas) on which an inference operation C is defined," three approaches are distinguished:

1. *A-scheme*: to pick a subset of the language, and use each element of the subset as a representation of absurdity,
2. *C-scheme*: to relate contradictions to inference, stating that inconsistency arises when all formulas are inferred,
3. *N-scheme*: to capture contradictions through an auxiliary notion of negation ($A, \neg A$ or $A \wedge \neg A$ if conjunction is available).

[1] See http://www.csus.edu/indiv/d/dowdenb/misc/inconsistency.htm.

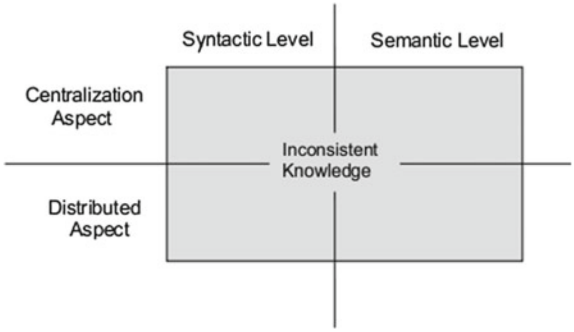

Fig. 1 Aspects and levels of inconsistency [38]

Inconsistency may be also represented in a form of *conflict profiles* as explained in [38]. Finally, inconsistency may be incorporated into formal logic, for instance as in multi-valued logics [9] in which one can represent a statement that is both inferred to be true and false.

3.3 Aspects and Levels of Inconsistency

One can consider different aspects and levels of inconsistency. In [38], two-dimensional classification of inconsistency is given as (1) syntactic vs. semantic level of inconsistency, and (2) centralization vs. distributed aspect of it (see Fig. 1). Inconsistency may be considered in *distribution aspect* where the basic cause of inconsistency is the independence of knowledge agents or knowledge processing mechanisms, or in *centralization aspect* where inconsistency is caused by the dynamic change of the world. It can also be identified and processed on a *syntactic* and *semantic* level. Alternatively, a distinction given in [26] states that inconsistency can be checked for in a purely *logical* way (e.g., p and $\neg p$ are present in the knowledge under discourse), or as *material* inconsistency, when two pieces of knowledge are invalid together due to the assumed interpretation.

3.4 Measuring Inconsistency

Binary distinction between consistent and inconsistent knowledge base is often insufficient. In order to better understand inconsistency and apply appropriate technique to handle it, one should recognize degree of inconsistency measured in some dimensions. Several methods, models, and metrics have been proposed to measure inconsistency [18, 21, 30]. Classifications for inconsistent theories have been proposed in [17]. In [22], different dimensions of measuring inconsistency are explained, selected approaches are compared, and their applications in cases as negotiation between agents or comparing heterogeneous sources are presented. Finally, inconsistency measures can be used to decide, if resolving inconsistency is worthwhile, taking into consideration associated loss of information [19].

3.5 Selected Approaches to Inconsistency

Actual and potential contradictions In [11], two approaches are distinguished, which lay the ground for various methods of inconsistency handling. *Actual contradictions* view assumes that contradictions appear naturally and thus a representation of them within a formalism and reasoning mechanisms in the presence of them should be provided. In *potential contradictions* view, it is claimed that in reality contradictions do not appear, and when contradicted information is given, there are some statements 'responsible' for it. Reasoning then conforms to trying to identify the 'strongest argument,' where all arguments that oppose it are discarded. This approach assumes some mechanism to resolve conflicts and obviate the potential of any contradiction. Practical implementations of this view are, e.g., defeasible reasoning, or modal formulation of default logic.

Methods for Handling Inconsistency Following different aspects and levels of inconsistency, numerous methods for handling it have been developed (survey of them is beyond the scope of this paper, for reference see e.g., [10, 11, 38]). Ordering them from most restrictive to most tolerating inconsistency, one can

1. Discard inconsistency, e.g., by removing/forgetting inconsistent statements.
2. Suppress it, by selecting consistent subsets (removing union of the minimally inconsistent subsets or taking intersection of the maximally consistent subsets, or removing smallest number of assumptions), isolating or repairing.
3. Do not accept it but use to find a consensus, or in argumentation frameworks.
4. Tolerate or accept it, by representing and reasoning with it (paraconsistency).
5. Accept and use constructively, e.g., learn from inconsistency [23].

Amending Classical Logic for Handling Inconsistency Because classical logic collapses in the presence of inconsistency, several amendments have been proposed [11]:

1. Syntactic-based approach: fragment of classical proof theory that is sufficiently weakened so that an arbitrary B fails to be inferred from $A \wedge \neg A$ (numerous paraconsistent logics [41]).
2. Semantic-based approach: conjoining the truth values *false* and *true* can be meaningful and results in a new truth values, namely *contradictory* (three-valued logic, Belnap's four-valued logic [9]).
3. Restricting the use of certain classical proof rules so as to avoid the proof of an arbitrary B following from $A \wedge \neg A$.
4. If premises form contradictory statements, then focusing on consistent parts of this collection and reasoning only from such consistent parts (subsets).

Paraconsistent Logics Main assumption of paraconsistent logics is the rejection of inference explosion (ECQ) in the presence of inconsistency. This shared assumption may be realized by different means. Selected paraconsistent logics include discussive logic, non-adjunctive systems, preservation, adaptive logics, logics of formal inconsistency, many-valued logics, or relevant logics (for a discussion and further reference see [41]). Paraconsistent logic has significant overlap with many-valued logic; however, not all paraconsistent logics are many-valued (and not all many-valued logics are paraconsistent). Dialetheic logics, which are also many-valued, are paraconsistent, but the converse does not hold. Intuitionistic logic allows $A \vee \neg A$ not to be equivalent to true, while paraconsistent logic allows $A \wedge \neg A$ not to be equivalent to false. Thus paraconsistent logic can be regarded as the "dual" of intuitionistic logic.

3.6 Selected Application Areas

Inconsistency has been widely studied in the areas of software engineering [39], legal knowledge engineering, verification of knowledge-based systems [25], diagnostics [27], and artificial intelligence (i.e., robotics, knowledge representation, and reasoning). With the advent of the Semantic Web, numerous proposals have been discussed for semantic knowledge management, both on the levels of semantic annotations in RDF/S [4] as well as ontologies [16, 20, 29, 38] (for a discussion about inconsistency handling in Semantic Web environments see [2]). Recently, inconsistency management has also been studied in multi-context systems [14] and traffic regulations within smart cities [8].

4 Interpretation of Inconsistency in Semantic Wikis

Based on practical experience and previous work [1, 34], we introduce the notion of *Collaborative Knowledge Authoring* in semantic wiki systems. It is a process of eliciting, structuring, formalizing, and operationalizing knowledge performed by conscious agents that collaborate on a knowledge level toward achieving a common goal, can communicate and help each other, as well as change their opinions. One can see that we highlight the following characteristics:

A Collaborative:

- multiple *conscious* agents (authors, knowledge engineers),
- communication, help and dynamics of opinions among agents.

Knowledge:

- consideration on a knowledge level [37], and consequently various methods of knowledge representation and reasoning (hybrid representations, different levels of formality [6])

Authoring:

- set of tasks (knowledge elicitation, structuring, formalization, and operationalization)
- common goal: development of a useful artifact

The developed knowledge base should be *valuable* for its users. This user-centric perspective intuitively alters the quality criteria of the system. Specifically, one should be able to *meaningfully* answer queries posed to the system. This does not necessarily require sustaining consistency.

Inconsistency in such a semantic wiki environment may be considered as driven by the following three factors: distributed knowledge authoring, dynamic change of the system, and hybrid knowledge representation. Analyzing the origins of inconsistency, one can distinguish the following categories:

1. Distributed knowledge authoring
   (a) independent knowledge sources/authors
   (b) independent knowledge processing
   (c) unconscious disagreement
      - in/competence of experts
      - inaccuracy of statements
   (d) conscious disagreement
      - different opinions
      - different conceptualizations
2. Dynamic change of the system
   (a) revisions of knowledge bases
   (b) assertion of facts inconsistent with existing KB
   (c) assertion of rules making the KB inconsistent
   (d) dynamic changes of the world (making the KB outdated)
3. Knowledge representation
   (a) incompatibilities between models expressed with different KR
      - e.g., incorrect combination of disjoint and derives relations
   (b) improper conceptualizations
      - polysemy (missing disambiguation of different word senses)
      - overgeneralized concepts

Moreover, we distinguish the following types of inconsistency:

- Syntactic inconsistency within a model in chosen representation.

- Semantic inconsistency between knowledge represented in different formalisms, possibly of different levels of formality [6, 7].
- Material inconsistency: if a behavior of the system is modeled within the KBS and the model is inconsistent with the actual execution of it.

Impact of the inconsistency depends on a formalization level of the knowledge base (the more the formal representation, the more significant is the inconsistency).

5 On Possible Usefulness of Inconsistency

Inconsistency can be acceptable, or even desirable in a system, as long as it has appropriate mechanisms for acting on it [15]. Sometimes, systematic eradication of contradictions is a mistake on several grounds [11]. First one is pragmatic: they may be hard to detect. Second, they can be more informative than any consistent revision of the theory (for a discussion see [15]).

While inconsistencies are undesired in such situations as specification of a plan or sensor fusion in robotics, they proved to be useful in, e.g., law (inconsistencies in opposition case), income tax database, where contradictions point to fraud investigation (contradictory information should be then kept and reasoned with), or in preliminary stage of software engineering (requirements capture stage), where premature resolution can force an arbitrary decision to be made without the choice being properly considered.

Inconsistency may be a useful trigger for logical actions, e.g., in directing reasoning, instigating the natural processes of argumentation, information seeking, interaction, knowledge acquisition and refinement, adaptation, and learning [23].

Tolerating inconsistency in a semantic wiki system is *pragmatic*, because inconsistency is hard to detect if hybrid representation is considered. Moreover, consistency is *not necessary for the knowledge base to be valuable*, if the quality is ensured collaboratively by employing social mechanisms such as voting, discussions, or "likes." During divergent thinking collaborative tasks, such as brainstorming sessions in research collaboration [10] or early prototyping while developing innovative ideas [40], inconsistency may inspire new associations and lead to more interesting solutions. Furthermore, inconsistency of opinions is thought-provoking and valuable in some applications (e.g., recommendation systems). Finally, inconsistency may be helpful in discovering potential areas of improvement, and be a trigger to acquire more knowledge.

6 Exemplary Use Case

A recommendation system has been implemented in Loki [33–35], a semantic wiki that supports semantic annotations and rules. Knowledge base about movies has been developed collaboratively by editing semantic wiki pages enhanced with semantic annotations (see Fig. 2).

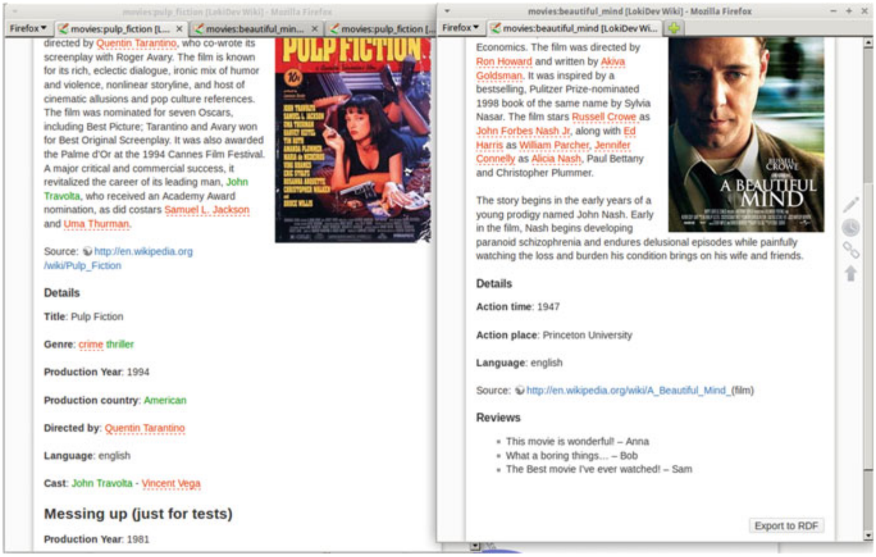

Fig. 2 Loki semantic wiki system: movies use case

Semantic annotations are mapped to the underlying logical representation that allows to reason automatically. Information may be obtained from wiki by posing semantic queries in *ask* and *SPARQL* languages, as well as Prolog goals. The system can also give recommendations based on rules defined for a user [1]. Information may be exported to RDF/XML.

Inconsistency arises when multiple users edit wiki pages and give contradictory information about the same objects. In order to easily locate and asses inconsistency, a visualization plugin has been developed.[2] The plugin highlights inconsistencies for whole namespaces (see Fig. 3) and single pages (see Fig. 4).

Whether a contradiction is acceptable or not depends on the ontology to which the system conforms. For instance, there are *functional* relations and attributes, such as `title` of a movie or its `production year`. For these properties, two different values constitute undesired inconsistency. On the other hand, for properties such as `rating`, several different values may be given and in this case, the inconsistency is natural and represents various opinions of the users. Thus Loki visualization plugin provides a simple configuration mechanism to define what properties are appropriate for given classes of objects and which of them are functional (see Fig. 4). Consequently, not only inconsistencies can be easily identified, but they also may be assessed and treated appropriately.

[2]A prototype implementation of the plugin was carried out by master students Magdalena Chmielewska and Tomasz Szczęśniak.

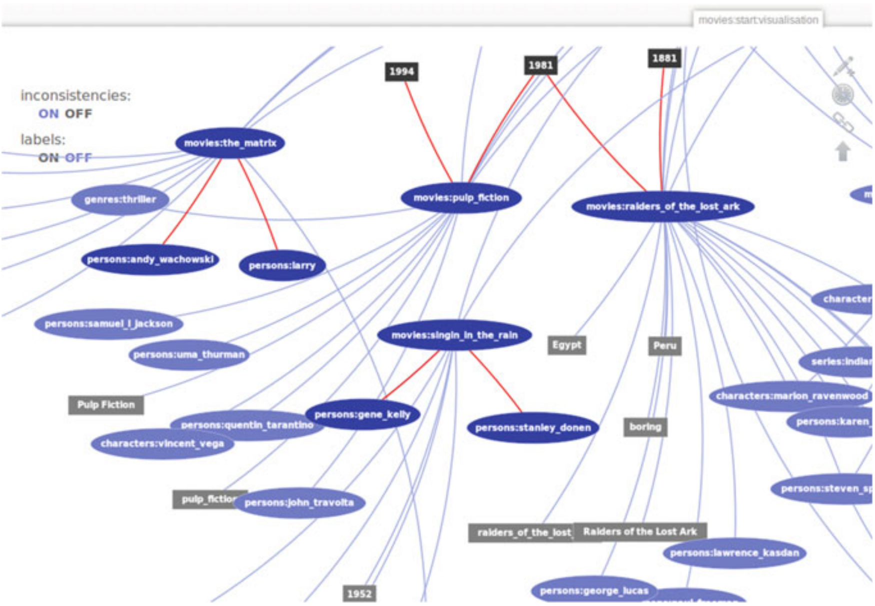

Fig. 3 Loki inconsistency visualization plugin: movies namespace

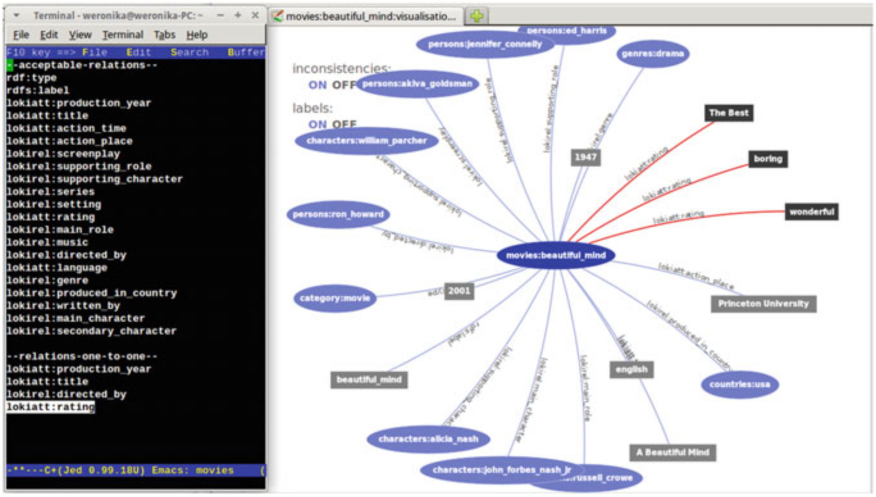

Fig. 4 Identified inconsistency highlighted for analysis

Recognizing more types of inconsistency is planned to be implemented in future. Further refinement of the system may be considered by employing voting or consensus methods to determine appropriate version of information.

7 Summary and Outlook

Inconsistency is an inherent phenomenon in collaborative knowledge authoring. We claim that only after a thorough analysis, one can take a reasonable decision how to handle identified inconsistency. Particularly, it is not obvious that inconsistency must be considered unacceptable. It may be useful and in this paper, we showed an exemplary use case. For future work, we will analyze how to adapt the following techniques for our semantic wiki environment: consensus methods [38], paraconsistent logics [41], 4-valued logics [9, 29], argumentation frameworks [16], belief revision, updating knowledge (e.g., ontologies) [24], inconsistent subsets, and semantic relevance metrics [20]. In fact our environment may be extended by a set of plugins simplifying the collaborative knowledge authoring that considers inconsistency as a useful phenomenon.

References

1. Adrian, W.T., Bobek, S., Nalepa, G.J., Kaczor, K., Kluza, K.: How to reason by HeaRT in a semantic knowledge-based wiki. In: Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2011, pp. 438–441. Boca Raton, Florida, USA, Nov 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6103361&tag=1
2. Adrian, W.T., Ligęza, A., Nalepa, G.J.: Inconsistency handling in collaborative knowledge management. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the Federated Conference on Computer Science and Information Systems—FedCSIS 2013, pp. 1261–1266. IEEE, Krakow, Poland, 8–11 Sept 2013
3. Adrian, W.T., Nalepa, G.J., Ligęza, A.: On potential usefulness of inconsistency in collaborative knowledge engineering. In: Proceedings of the 8th International Conference on Knowledge, Information and Creativity Support Systems (2013)
4. Analyti, A., Antoniou, G., Damásio, C.V., Wagner, G.: Extended rdf as a semantic foundation of rule markup languages. J. Artif. Intell. Res. **32**(1), 37–94 (2008)
5. Baumeister, J., Nalepa, G.J.: Verification of distributed knowledge in semantic knowledge wikis. In: Lane, H.C., Guesgen, H.W. (eds.) FLAIRS-22: Proceedings of the Twenty-Second International Florida Artificial Intelligence Research Society conference: 19–21 May 2009, Sanibel Island, Florida, USA, pp. 384–389. FLAIRS, AAAI Press, Menlo Park, California (2009), to be published
6. Baumeister, J., Reutelshoefer, J., Puppe, F.: Engineering intelligent systems on the knowledge formalization continuum. Int. J. Appl. Math. Comput. Sci. (AMCS) **21**(1) (2011). http://ki.informatik.uni-wuerzburg.de/papers/baumeister/2011/2011-Baumeister-KFC-AMCS.pdf
7. Baumeister, J., Seipel, D.: Anomalies in ontologies with rules. Web Semant. Sci. Serv. Agents World Wide Web **8**(1), 55–68 (2010). http://www.sciencedirect.com/science/article/pii/S1570826809000778
8. Beck, H., Eiter, T., Krennwallner, T.: Inconsistency management for traffic regulations: formalization and complexity results. In: del Cerro, L.F., Herzig, A., Mengin, J. (eds.) Logics in Artificial Intelligence. Lecture Notes in Computer Science, vol. 7519, pp. 80–93. Springer, Heidelberg (2012). http://dx.doi.org/10.1007/978-3-642-33353-8_7
9. Belnap Jr, N.D.: A useful four-valued logic. In: Modern Uses of Multiple-valued Logic, pp. 5–37. Springer (1977)
10. Bertossi, L., Hunter, A., Schaub, T.: Introduction to inconsistency tolerance. In: Bertossi, L., Hunter, A., Schaub, T. (eds.) Inconsistency Tolerance. Lecture Notes in Computer Sci-

ence, vol. 3300, pp. 1–14. Springer, Heidelberg (2005). http://dx.doi.org/10.1007/978-3-540-30597-2_1
11. Besnard, P., Hunter, A.: Introduction to actual and potentialcontradictions.In: Besnard, P., Hunter, A. (eds.) Reasoning with Actual and PotentialContradictions, Handbook of Defeasible Reasoning and Uncertainty ManagementSystems, vol. 2, pp. 1–9. Springer Netherlands (1998). http://dx.doi.org/10.1007/978-94-017-1739-7_1
12. Bienvenu, M., Rosati, R.: New inconsistency-tolerant semantics for robust ontology-based data access. In: Description Logics, pp. 53–64 (2013)
13. Coenen, F., Bench-Capon, T., Boswell, R., Dibie-Barthélemy, J., Eaglestone, B., Gerrits, R., Grégoire, E., Ligęza, A., Laita, L., Owoc, M., Sellini, F., Spreeuwenberg, S., Vanthienen, J., Vermesan, A., Wiratunga, N.: Validation and verification of knowledge-based systems: report on eurovav99. Knowl. Eng. Rev. **15**(2), 187–196, June 2000. http://dx.doi.org/10.1017/S0269888900002010
14. Eiter, T., Fink, M., Weinzierl, A.: Preference-based inconsistency assessment in multi-context systems. In: Janhunen, T., Niemelä, I. (eds.) Logics in Artificial Intelligence. Lecture Notes in Computer Science, vol. 6341, pp. 143–155. Springer, Berlin (2010). http://dx.doi.org/10.1007/978-3-642-15675-5_14
15. Gabbay, D., Hunter, A.: Making inconsistency respectable: a logical framework for inconsistency in reasoning, part i—a position paper. In: Jorrand, P., Kelemen, J. (eds.) Fundamentals of Artificial Intelligence Research, Lecture Notes in Computer Science, vol. 535, pp. 19–32. Springer, Berlin (1991). http://dx.doi.org/10.1007/3-540-54507-7_3
16. Gomez, S.A., Chesnevar, C.I., Simari, G.R.: Reasoning with inconsistent ontologies through argumentation. Appl. Artif. Intell. **24**(1–2), 102–148 (2010)
17. Grant, J.: Classifications for inconsistent theories. Notre Dame J. Form. Log. **19**(3), 435–444 (1978)
18. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. J. Intell. Inf. Syst. **27**(2), 159–184 (2006)
19. Grant, J., Hunter, A.: Measuring consistency gain and information loss in stepwise inconsistency resolution. In: Liu, W. (ed.) Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Lecture Notes in Computer Science, vol. 6717, pp. 362–373. Springer, Berlin (2011)
20. Huang, Z., Harmelen, F.: Using semantic distances for reasoning withinconsistent ontologies. In: Proceedings of the 7th International Conferenceon The Semantic Web, pp. 178–194. ISWC '08, Springer-Verlag, Berlin (2008)
21. Hunter, A.: How to act on inconsistent news: ignore, resolve, or reject. Data Knowl. Eng. **57**(3), 221–239 (2006). http://dx.doi.org/10.1016/j.datak.2005.04.005
22. Hunter, A., Konieczny, S.: Approaches to measuring inconsistentinformation. In: Bertossi, L., Hunter, A., Schaub, T. (eds.) Inconsistency Tolerance,Lecture Notes in Computer Science, vol. 3300, pp. 191–236. Springer, Berlin (2005). http://dx.doi.org/10.1007/978-3-540-30597-2_7
23. Kühnberger, K.U., Geibel, P., Gust, H., Krumnack, U., Ovchinnikova, E., Schwering, A., Wandmacher, T.: Learning from inconsistencies in an integrated cognitive architecture. In: Proceedings of the 2008 conference on Artificial General Intelligence 2008, pp. 212–223. IOS Press, Amsterdam, The Netherlands, The Netherlands (2008). http://dl.acm.org/citation.cfm?id=1566174.1566194
24. Lenzerini, M., Savo, D.F.: Updating inconsistent description logic knowledge bases. In: ECAI. pp. 516–521 (2012)
25. Ligęza, A.: Intelligent data and knowledge analysis and verification; towards a taxonomy of specific problems. In: Vermesan, A., Coenen, F. (eds.) Validation and Verification of Knowledge Based Systems, pp. 313–325. Springer, US (1999). http://dx.doi.org/10.1007/978-1-4757-6916-6_21
26. Ligęza, A.: A 3-valued logic for diagnostic applications. In: Yannick, P., Alexander Feldman, A.G. (ed.) Diagnostic REAsoning: Model Analysis and Performance, August, 27th, Montpellier, France (2012). http://dreamap.sciencesconf.org/conference/dreamap/eda_en.pdf

27. Ligęza, A., Kościelny, J.M.: A new approach to multiple fault diagnosis: a combination of diagnostic matrices, graphs, algebraic and rule-based models. the case of two-layer models. Appl. Math. Comput. Sci. **18**(4), 465–476 (2008)
28. Ligęza, A., Nalepa, G.J.: A study of methodological issues in design and development of rule-based systems: proposal of a new approach. Wiley Interdisc. Rev.: Data Min. Knowl. Discovery **1**(2), 117–137 (2011)
29. Ma, Y., Hitzler, P.: Paraconsistent reasoning for OWL 2. In: Polleres, A., Swift, T. (eds.) Web Reasoning and Rule Systems, Lecture Notes in Computer Science, vol. 5837, pp. 197–211. Springer, Berlin (2009)
30. Ma, Y., Qi, G., Hitzler, P.: Computing inconsistency measure based on paraconsistent semantics. J. Logic Comput. **21**(6), 1257–1281 (2011)
31. Nalepa, G., Bobek, S., Ligęza, A., Kaczor, K.: Algorithms forruleinference in modularized rule bases. In: Bassiliades, N., Governatori, G.,Paschke, A. (eds.) Rule-Based Reasoning, Programming, and Applications. Lecture Notes in Computer Science, vol. 6826, pp. 305–312. Springer, Berlin (2011)
32. Nalepa, G., Bobek, S., Ligęza, A., Kaczor, K.: HalVA—ruleanalysisframework for XTT2 rules. In: Bassiliades, N., Governatori, G., Paschke, A.(eds.) Rule-Based Reasoning, Programming, and Applications. Lecture Notes inComputer Science, vol. 6826, pp. 337–344. Springer, Berlin (2011). http://www.springerlink.com/content/c276374nh9682jm6/
33. Nalepa, G.J.: PlWiki—a generic semantic wiki architecture. In: Nguyen, N.T., Kowalczyk, R., Chen, S.M. (eds.) Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, First InternationalConference, ICCCI 2009, Wroclaw, Poland, October 5–7, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5796, pp. 345–356. Springer (2009)
34. Nalepa, G.J.: Collective knowledge engineering with semantic wikis. J. Univ. Comput. Sci. **16**(7), 1006–1023 (2010). http://www.jucs.org/jucs_16_7/collective_knowledge_engineering_with
35. Nalepa, G.J.: Loki—semantic wiki with logical knowledgerepresentation. In: Nguyen, N.T. (ed.) Transactions on Computational Collective Intelligence III, Lecture Notes in Computer Science, vol. 6560, pp. 96–114. Springer (2011). http://www.springerlink.com/content/y91w134g03344376/
36. Nalepa, G.J., Ligęza, A., Kaczor, K.: Formalization and modeling of rules using the XTT2 method. Int. J. Artif. Intell. Tools **20**(6), 1107–1125 (2011)
37. Newell, A.: The knowledge level. Artif. Intell. **18**(1), 87–127 (1982)
38. Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management (Advanced Information and Knowledge Processing). Springer, London (2008)
39. Nuseibeh, B., Easterbrook, S., Russo, A.: Making inconsistency respectable in software development. J. Syst. Softw. **58**(2), 171–180 (2001)
40. Pniewska, J., Adrian, W.T., Czerwoniec, A.: Prototyping: is it a more creative way for shaping ideas. In: Proceedings of the International Conference on Multimedia, Interaction, Design and Innovation. pp. 18:1–18:8. MIDI ’13, ACM, New York, NY, USA (2013). http://doi.acm.org/10.1145/2500342.2500361
41. Priest, G., Tanaka, K., Weber, Z.: Paraconsistent logic. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Summer 2013 edn. (2013)

Springer

http://www.springer.com/978-3-319-19089-1

Knowledge, Information and Creativity Support Systems:
Recent Trends, Advances and Solutions
Selected Papers from KICSS'2013 - 8th International
Conference on Knowledge, Information, and Creativity
Support Systems, November 7-9, 2013, Kraków, Poland
Skulimowski, A.M.J.; Kacprzyk, J. (Eds.)
2016, VIII, 566 p. 146 illus., 11 illus. in color., Softcover
ISBN: 978-3-319-19089-1