On the Practicality of Viewpoint-Based Requirements Engineering*

Tim Menzies[1], Sam Waugh[2]

[1] AI Department, Computer Science and Engineering, University of NSW, Australia, tim@menzies.com;
[2] Defence Science and Technology Organisation, Air Operations Division, Melbourne, Australia, sam.waugh@dsto.defence.gov.au

Abstract. Requirements engineering is often characterised as the management of conflicts between the viewpoints of different stakeholders. This approach is only useful if there is some benefit in moving a specification from one viewpoint to another. In this study, the value of different viewpoints was assessed using a range of different models (ranging from correct to very incorrect), different fanouts, different amounts of data available from the domain, and different temporal linking policies. In all those models, no significant difference was observed between viewpoints.

1 Introduction

Acquiring and consolidating software requirements from different stakeholders is a time-consuming and costly process. If these different viewpoints are poorly managed, the specifications have to be repeatedly reworked or the runtime system has to be extensively modified [9]. Viewpoint-based requirements engineering researchers characterise this process as the management of conflicts between the viewpoints of different stakeholders (e.g. [7,9,18]). Viewpoint-based requirements engineering (hereafter V-RE) assumes that there is some benefit in managing more than one viewpoint. Can we check that assumption?

A core function in viewpoint management is assessing the relative merits of different positions. The limits to this assessment process are the limits to V-RE. One framework for assessing competing options is abduction. Informally, abduction is the inference to the best explanation [19]. More precisely, abduction makes assumptions in order to complete some inference. Mutually exclusive assumptions are managed in separate worlds [17]. That is, given a theory containing contradictions, abduction sorts those contradictions into consistent portions. If the theory is the union of the ideas from different stakeholders, then abduction becomes an V-RE tool. Queries can be written to assess the different worlds. V-RE negotiation then becomes a discussion of the trade-offs between different worlds (an example of this process is given below). Once V-RE has been mapped into abduction, then the limits to V-RE can be found by exploring the limits to abduction. This article will explore those limits as follows. Firstly, an example will show the mapping from V-RE to abduction. Hence, we can show that

* Pacific Rim Internalional Conference on AI, Singapore, Nov 22-27 1998

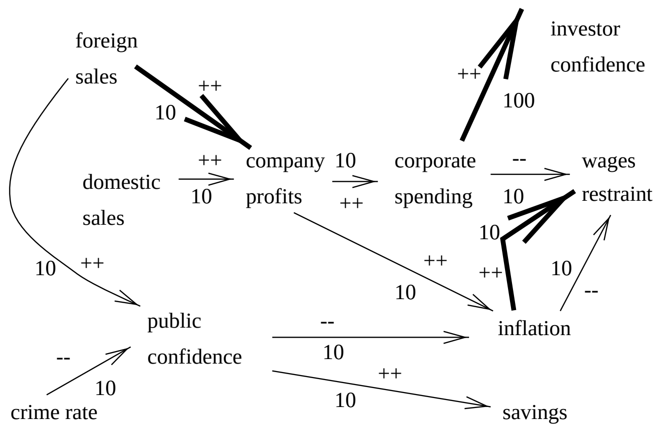

Fig. 1. A model from two experts.

V-RE is NP-hard. Secondly, an experiment will be defined to find the limits to that abductive process. Thirdly, the results of that experiment will show that, at least in the domain studied, different viewpoints are rare and that there is little benefit in moving between them.

2 Example of Abduction

This section offers an example of abductive-based V-RE. Consider the theory shown in Figure 1

This figure is written in the QCM language [17] by two economists: Dr. Thick and Dr. Thin. In QCM, theory variables have three states: *up*, *down* or *steady*. These values model the sign of the first derivative of these variables and model the rate of change in each value. Dependencies between them can be created as follows. The direct connection between *foreignSales* and *companyProfits* (denoted with plus signs) means that *companyProfits* being *up* or *down* should be connected back to *foreignSales* being *up* or *down* respectively. The inverse connection between *publicConfidence* and *inflation* (denoted with minus signs) means that *inflation* being *up* or *down* should be connected back to *publicConfidence* being *down* or *up* respectively. We assume that, somehow, we have knowledge of the relative costs of each inference step in the model: each edge is the model is annotated with its numeric weight. Dr. Thick's and Dr. Thin's ideas are shown in thick and thin lines respectively. Note that our doctors disagree on the connection between *inflation* and *wagesRestraint*.

How can we test if Dr. Thick or Dr. Thin are saying anything sensible? One method is to use a library of known or desired behaviour. Dr. Thick or Dr. Thin's ideas are sensible if they can reproduce that behaviour. Further, one expert's theory is better than the other theory if that if that theory can explain more known behaviour than its competitors.

This method has at least two problems. Firstly, it may be artificial to demand that (e.g.) Dr. Thick is totally correct and Dr. Thin is totally wrong. A more sensible approach may be to combine portions of Dr. Thick and Dr. Thin's knowledge in order to perform some useful task. Secondly, V-RE researchers such as Easterbrook [6], Finkelstein [9], and Nuseibeh [18] argue that we should routinely expect specifications to reflect different and inconsistent viewpoints. In classical deductive logic, if we can prove a contradiction in a theory, then that theory becomes useless since anything at all can be inferred from that contradiction. Consider the case of *(foreignSales=up, domesticSales=down)* being inputs to the above economics theory. We can now infer two contradictory conclusions: *companyProfits=up* and *companyProfits=down*. In classical deductive logic, we would have to declare our economics theory useless.

A better approach for checking on Dr. Thick and Dr. Thin is graph-based abductive validation [15, 17]. Graph-based abductive validation builds explanations (worlds) for each pair of inputs-outputs in the library of known behaviour. Worlds are built by finding all possible proofs from outputs back to inputs across a directed graph like our economics model. Each maximal consistent subset of those proofs is a world. Worlds are internally consistent. Contradictory assumptions are stored in separate worlds. Each world is scored via its intersection with the total number of outputs we are trying to explain. A theory is then assessed by computing the largest score of its worlds.

This approach to testing was first proposed by Feldman and Compton [8], then generalised and optimised by Menzies [15, 17]. Abductive validation has found a large number of previously unseen errors in models taken from international refereed scientific publications. The errors had not previously been detected and has escaped international peer review prior to publication. To see how graph-based abductive validation contributes to V-RE, consider our economics theory and the case where the inputs are *(foreignSales=up, domesticSales=down)* and the outputs are *(investorConfidence=up, inflation=down, wageRestraint=up)*. The six proofs `P` which can connect inputs to outputs are:

- `P.1`: *foreignSales=up, companyProfits=up, corporateSpending=up, investorConfidence=up.*
- `P.2`: *domesticSales=down, companyProfits=down, corporateSpending=down wageRestraint=up.*
- `P.3`: *domesticSales=down, companyProfits=down,inflation=down.*
- `P.4`: *domesticSales=down, companyProfits=down, inflation=down, wagesRestraint=up.*
- `P.5`: *foreignSales=up, publicConfidence=up, inflation=down.*
- `P.6`: *foreignSales=up, publicConfidence=up, inflation=down, wageRestraint=up.*

Note that these proofs contain contradictory assumptions; e.g. *corporateSpending=up,* in `P.1` and *corporateSpending=down* in `P.2`. When we sort these proofs into maximal subsets that contain no contradictory assumptions, we arrive at the worlds shown in Figure 2.

Note that world one covers all our output goals while world two only covers two-thirds of our outputs.

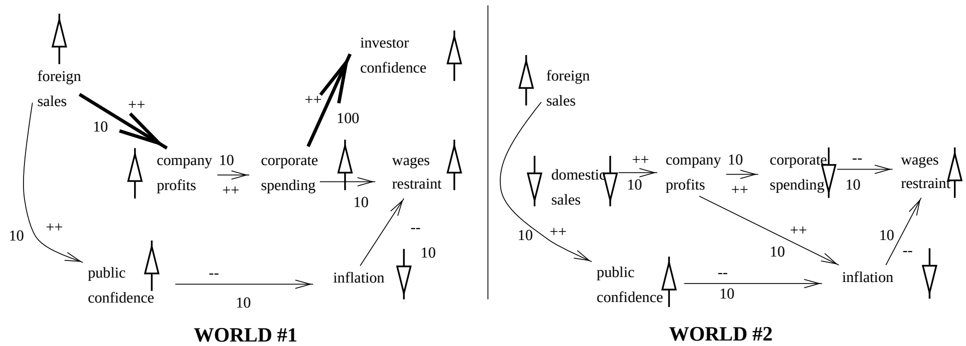

Fig. 2. Worlds from Figure 1.

V-RE is about facilitating a discussion, not automatically jumping to the next version of the specification. Hence, this abductive approach to V-RE does not offer automatic support for combining the ideas of different experts. However, it does support the automatic generation of reports describing the relative merits of the ideas of Dr. Thick and Dr. Thin. For example, if we combine portions of the ideas from Dr. Thick and Dr. Thin, then we can explain all known behaviour (see world one). Also, Dr. Thin's edges can be found in two worlds; i.e. with respect to the task of inputs *(foreignSales=up, domesticSales=down)* and outputs *(investorConfidence=up, inflation=down, wageRestraint=up)*, a single author's opinions are inconsistent. That is, there is some reason for doubting the view of Dr. Thin. Lastly, we have some evidence that we should endorse the views of Dr. Thin over Dr. Thick since Dr. Thin's ideas are cheaper than Dr. Thick. Consider the cost of world one which can support *investorConfidence=up*. This world contains the very expensive inference proposed by Dr. Thick. If we endorse only Dr. Thin, we get cheaper worlds but lose coverage of all outputs. Such a pragmatic trade off between cost and coverage could inform many debates over conflict resolution.

This abductive approach has technical advantages over other approaches to conflict detection and resolution. Firstly, Easterbrook [6] and Finkelstein [9] require that users enter their requirements into explicitly labeled separate viewpoints. Each viewpoint are assumed to be internally consistent. We have no need for this restrictive (and possibly overly-optimistic) assumption. Recalling the above example, abduction can handle inconsistencies within the opinions of a single user. Further, this approach can check if the explicitly labelled viewpoints are really different: if they don't generate different worlds when they are combined, then they are not truly different.

Secondly, this approach does not demand that we declare (e.g.) Dr. Thick is totally correct and Dr. Thin is totally wrong. This approach can find composite consistent statements that use portions of each expert's knowledge to solve some task (see world one, above).

Thirdly, graph-based abductive validation is not the JTMS-style [5] approach used in other conflict recognition and management systems (e.g. [23]). A JTMS searches for a single set of beliefs. Hence, by definition, a JTMS can only represent a single viewpoint at any one time. This approach is more like the ATMS [4] than a JTMS. An ATMS maintains all consistent belief sets. We believe that an ATMS approach is better suited to V-RE conflict management since the different belief sets are available for reflection.

Fourthly, one striking feature of other systems that support multiple-worlds (e.g. CAKE [23],TELOS [20]) is their implementation complexity. Rich and Waters especially comment on the complexity of their heterogenous architecture [23]. We have found that it is easier to build efficient implementations [15,16] using the above graph-based approach than using purely logical approaches (e.g. [11]). These tools do not suffer from the restrictions of other tools. For example, while Easterbrook's SYNOPTIC tool only permits comparisons of two viewpoints [6] (p113), our approach can compare N viewpoints.

Fifthly, the inference procedure described here avoids spurious state assignments. The state assignments proposed by a reasoner are its *envisionments.* Total envisionments are those behaviours which are possible, given some fixed collection of objects in some configuration. Extension generation in default logic [22] systems or the ATMS [4] produce total envisionments. A reasonable restriction on the total envisionments are the attainable envisionments; i.e. all behaviours possible from some given initial state. The QSIM qualitative reasoner uses attainable envisionments [13]. Graph-based abductive validation only finds the relevant envisionments; i.e. state assignments which can lead from inputs to outputs. Relevant envisionments answers the question: *Given some behaviour of interest can these behaviours be reached given certain state assignments?* To answer this question with total or attainable envisionments, one must compute the total or attainable envisionments, then search them for the required behaviour. This approach runs the risk of generating many behaviours that are irrelevant to the process of finding what percentage of known behaviours can be explained by a hypothetical model. For example, given the inputs and outputs of our above example, total envisionments would propose state assignments to *crimeRate* and attainable envisionments would propose state assignments to *savings*, even though these assignments are not relevant to reaching our output goals. To perform relevant envisionments, we restrict the search to the downstream transitive closure of the inputs and the upstream transitive closure of the outputs. For more details, see [17].

Sixthly, the simplicity of this approach can simplify an analysis of the limitations of V-RE (see below).

3 Limits to Abduction

The previous section argued that V-RE can be usefully expressed in an abductive framework. This section explores the computational limits of that abductive framework.

3.1 V-RE is NP-Hard

V-RE can be simply mapped into abduction and abduction is NP-hard. Selman and Levesque show that even when only one abductive explanation is required and the theory is restricted to be acyclic, then abduction is NP-hard [24]. Bylander et.al. make a similar pessimistic conclusion [2].

The specific graph-based abduction validation procedure discussed above is also NP-hard. That procedure grows proofs up from outputs back to inputs. As the proof grows, state assignments (e.g. *domesticSales=up*) is added to the proof. A proof must be consistent; i.e. it must not contain items that contradict other items in the proof. This proof invariant makes this procedure NP-hard. Gabow et.al. [10] showed that finding a directed path across a directed graph that has at most one of a set of forbidden pairs is NP-hard. Our forbidden pairs are assignments of different values to the same variable; e.g. the pairs *domesticSales=up* and *domesticSales=down*.

Pragmatic software engineers often build practical systems for problems that are theoretically NP-hard problems. Hence, merely showing that V-RE is NP-hard is not sufficient reason to abandon that approach. However, the experimental results discussed below are of more practical concern.

3.2 Looking for Multiple Viewpoints

A premise for V-RE is that different viewpoints exist. At first glance, this seems very likely. V-RE is abduction and Kakas et.al. [12] remark that a distinguishing feature of abduction is the generation of multiple explanations (a.k.a. worlds). Researchers into qualitative models (e.g. our economics theory) often comment on the indeterminacy of such models (the generation of too many worlds). Clancy and Kuipers say that qualitative indeterminacy is the major restriction to the widespread adoption of qualitative reasoners [3].

Curiously, and contrary to the experience of Clancy, Kuipers, Kakas, et.al, graph-based abductive validation exhibits very little indeterminacy [14]. That is, when we checked for multiple worlds (a.k.a. viewpoints), we could not find them. This was such a surprising observation that the following experiment was conducted. The aim of the experiment was to try and force graph-based abductive validation to generate numerous worlds.

Firstly, some quantitative equations of a fisheries system was taken from Bossel [1] (pages 135-141) and converted into the QCM-style diagram in Figure 3. Note the two variables *change in boatNumbers* and *change in fishPopulation*. These change variables explicitly model the time rate of change of variables. The simulation data from the quantitative equations offered state assignments at every year. To handle such temporal simulations, the qualitative model was copied, once for every time tick in the simulation. That is, variables like *fishCatch* were copied to become *fishCatch@1*, *fishCatch@2*, etc. Variables at time *i* were connected to variables at time *i+1* using a *temporal linking policy* (discussed below).

Once fisheries was copied, graph-based abductive validation was used to try and reproduce data sets generated from the quantitative equations. Fisheries is only one model. Conclusions drawn from the behaviour of one model are hardly general. Hence, we built several *mutators* to generate 100,000s of problems. The generated problems contained (i) a range of different models (ranging from correct to very incorrect); (ii) models with different fanouts, (iii) different amounts of data available from the domain; (iv) different temporal linking policies.

One mutator added edges to fisheries. Basic fisheries has 12 nodes and 17 edges (fanout=17/12=1.4). This mutator added 0, 5, 10, 15, 20, 25 or 30 new edges at random (checking all the time that the added edges did not exist already in the theory). That is, the model fanout was mutated from 1.4 to (17+30/12=3.9).

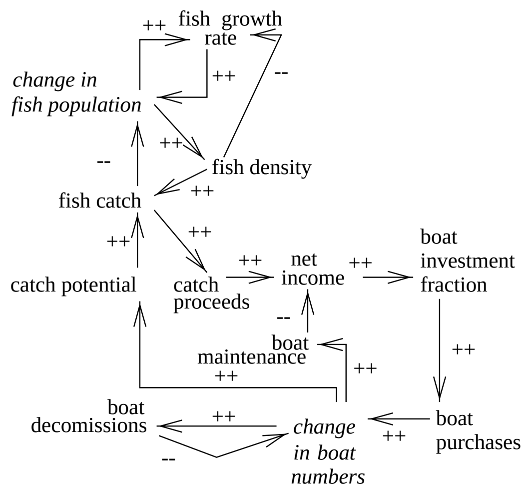

Fig. 3. The fisheries model. Adapted from [1] (pp135-141).

A second mutator corrupted the edges on the original fisheries model. This mutator select N links at random in the fisheries model and flipped the annotation (*++* to – and visa versa). There are 17 edges in the fisheries model. Note that as the number of edges mutated increases from 0 to 17, the mutated model becomes less and less like the original model. That is: at *mutations=0* we are processing the correct fisheries model; at *mutations=17* we are processing a very incorrect fisheries model; at *mutations=2..16* we are processing progressively worse fisheries models.

A third mutator changed the amount data available to graph-based abductive validation. The Bossel equations offered values for all variables at all time points. The third mutator threw away some of that data to produce data sets with 0,10,..,90 percent of the variables unmeasured (denoted as U percent unmeasured).

A fourth mutator changed how the variables were connected across time. The XNODE temporal linking policy connects all the explictedly-marked temporal variables from time i to time $i+1$; e.g. *change in boatNumbers=up@1* to *change in boatNumbers=up@2*. Note that there are only two explicit time variables in fisheries. It was thought that, since the number of connections were so few, this could artificially restrict world generation. Hence, another time linking policy was defined which made many cross-time links. The IEDGE temporal linking policy took all edges from A to B in the fisheries model and connected $A@i$ to $B@i{+}1$. XNODE and IEDGE are compared in the following example. Consider the theory *direct(A,B)* and *inverse(B,A)*. If we execute this theory over three time steps, then XNODE and IEDGE describe the search space illustrated in Figure 4.

The above mutators were combined as follows. The Bossel equations were used to generate 105 pairs of inputs and outputs. For statistical validity, the following procedure was repeated 20 times for each of IEDGE and XNODE:

- 0 to 17 edges were corrupted, once for each value of U (0,10,..,90). This lead to 7200 models (20*2*10*18) executed over the 105 input-output pairs (7200*105= 756,000 runs).
- 0, 5, 10, 15, 20, 25 or 30 edges were added, once for each value of U leading to 20*2*10=400 models being executed 105 times (42,000 runs)

The results are shown in Figure 5.

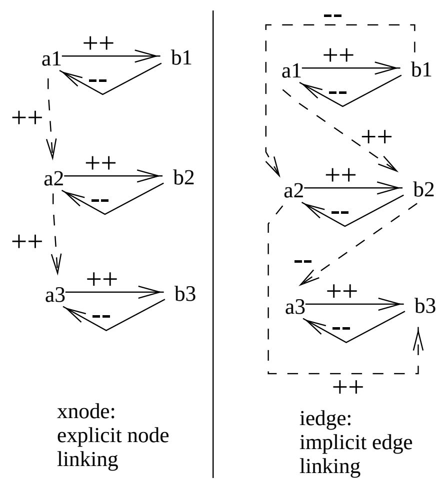

Fig. 4. *Direct(A,B)* and *inverse(B,A)* renamed over 3 time intervals using different time linking policies. Dashed lines indicate time traversal edges.

Note the low number of worlds generated. Our reading of the literature (e.g. [3, 12]) lead us the expect far more worlds than those observed here (maximum=5) Also, note the *hump* shape in all the results graphs. As we decrease the amount of data available, there is less information available to constrain indeterminacy. Hence, initially, less data means more worlds. However, after some point (around 50 percent unmeasured), another effect dominants and the number of worlds decreases. We conjecture that relevant envisionments are the cause of the low number of worlds. World-generation is a function of the number of conflicting assumptions made by the reasoner. As the percentage of unmeasured variables increases, the size of the input and output sets decreases. In total envisionments, this has no effect on the number of assumptions made since total envisionments offers assumptions for all variables. However, attainable envisionments make less assumptions while relevant envisionments make even less. Hence, for low-assumption envisionment policies (e.g. relevant envisionments), world-generation is reduced when the amount of data from the domain is reduced.

In summary. only certain interpretations of time (e.g. IEDGE) generate the multiple viewpoints needed for V-RE. How important are those worlds? In the

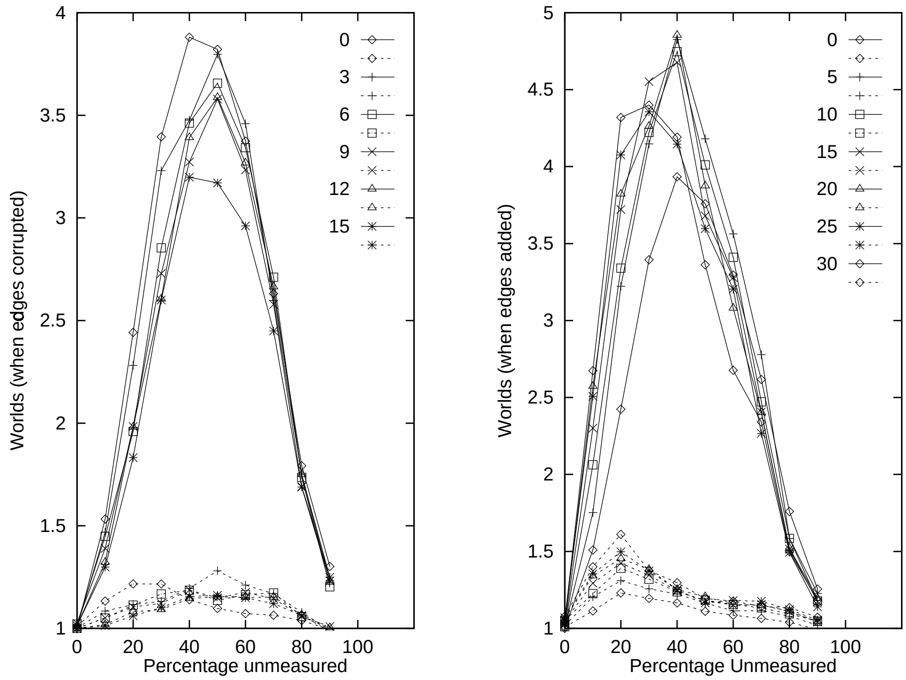

Fig. 5. IEDGE (solid lines), XNODE (dashed lines),

next experiment, we crippled the graph-based abductive validation device. Instead of returning the world(s) that explained the most number of outputs, we returned any single world, chosen at random. The results of that one-world abduction run was compared to the results gained from full multiple-world abduction. The test rig was the same as the edge corruption experiment described above; i.e. another 756,000 runs. A sample of those results are shown in Figure 6.

In these graphs, the percentage of outputs found in the worlds is shown on the y-axis (labelled *percent explicable*). For multiple-world abduction, the maximum percentage is shown; i.e. this is the most explanations that the theory can support. For one-world abduction, the percent of the one-world (chosen at random) is shown. Note that, at most, many-world reasoning was ten percent better than one-world reasoning (in the IEDGE graph for U=40 and 10 edges corrupted). The average improvement of many-world reasoning over one-world reasoning was 5.6 percent. That is, in millions of runs over thousands of models, there was very little difference seen in the worlds generated using one-world and multiple-world abduction.

4 Discussion

There are at least three limits to the above analysis. Firstly, it assumes a definition of the *worth* of a viewpoint along the lines of *what percent of known or*

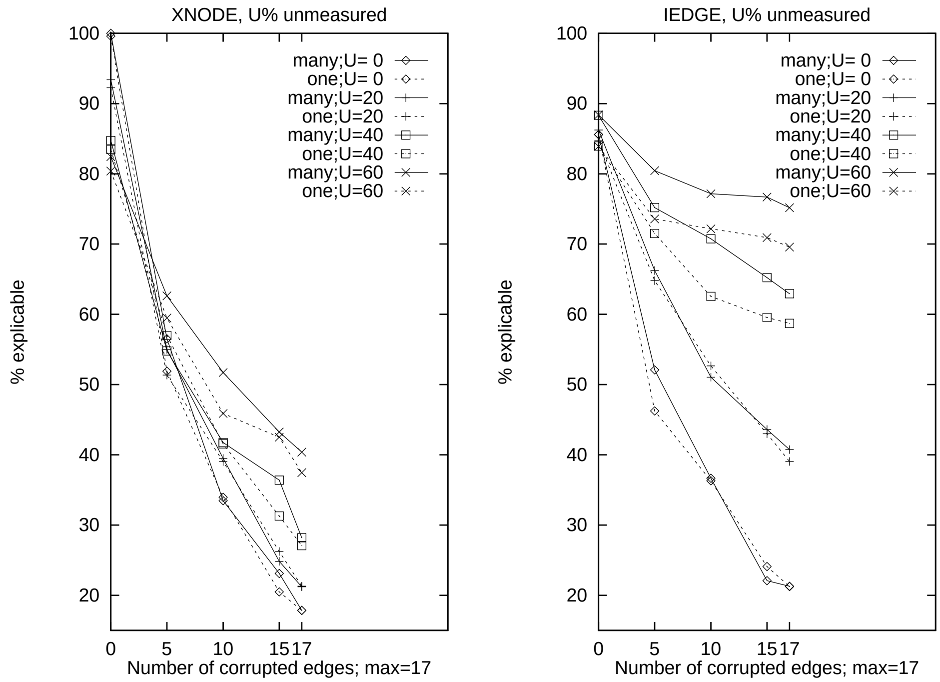

Fig. 6. Multiple-world abduction (solid line) vs one-world abduction (dashed line).

desired behaviour is found in that viewpoint?. Other definitions of testing exist; e.g. the syntactic anomaly detection work of Preece [21]. There is at least one advantage of our approach. If a theory cannot fully reproduce known or desired behaviour, then something *must* be wrong. The same cannot be said for other testing models. For example, Preece stresses that his work does not detect *errors*; rather it only detect *anomalies* which require further human investigation. That is, if a system fails a Preece-style check, it is still possible that nothing is really wrong.

Secondly, our scoring system for the worth of each viewpoint assigns the same score to a viewpoint explaining (e.g.) *a,b* as it does to a viewpoint explaining (e.g.) *c,d*. That is, the analysis here assumes a uniform distribution of goal *utilities* This is an incorrect assumption in domains where certain goals have very high utilities (compared to other goals). For example, two goals might be *healthy* and *well-dressed* and *healthy* might be more crucial than *well-dressed*. We considered experimenting with the effects of different utility distributions. However, that experiment was not conducted since we could not find guidance in the literature on what are reasonable utility distributions in real-world applications.

Thirdly, our analysis is based on mutations to fisheries: a single small theory. Perhaps an analysis of larger, more intricate theories, would offer different conclusions? While we acknowledge this possibility, we note fisheries was just the initial theory that seeded our mutators. Thousands of variants on fisheries were constructed, many of which were more complicated than fisheries (recall the first mutator added edges into the theory). As to larger theories, we showed above that V-RE is NP-hard; i.e. requirements engineering is necessarily limited to small theories. Our analysis of RE hence shares a size restriction with all other RE approaches. In support of this, we note that all the RE models we have ever seen have been small (but we have no data to support this informal observation).

Within the above limitations, we can make the following conclusions. V-RE is can only useful if (i) the viewpoints are truly different and (ii) there is some value in moving a specification from one viewpoint to another. After mapping V-RE to abduction, we have explored these two issues. Abduction can check if some explicitly named viewpoints are truly different: if they don't generate different worlds when they are combined, then they are not truly different. Also, by comparing one-world abductive validation to multiple-world abductive validation, we can assess the merit of exploring multiple viewpoints. Experimentally, we have shown here that for a range of problems (different models ranging from correct to very incorrect, different fanouts, different amounts of data available from the domain, different temporal linking policies) multiple-world reasoning can only generated marginally better results than one-world reasoning (ten percent or less). Hence, the domain explored by these experiments, there is no value in V-RE.

Should we then abandon V-RE? No: V-RE is useful in domains where truly different viewpoints (of significantly different value) can be generated. Alternatively, V-RE may also be of value in domains where increasing the value of a viewpoint by a few extra percent is of vital importance to the application. For example, in a medical domain, *a few percent* could imply saving thousands of deaths. However, what these results show is that even though a domain may *seem* to generate significantly different viewpoints, on average, these different viewpoints may not be worth considering. Multiple-world reasoners are hard to build and understand. Requirements engineers should carefully explore their domains before leaping to the complexity of multiple-world/viewpoint reasoners.

References

1. H. Bossel. *Modeling and Simulations.* A.K. Peters Ltd, 1994. ISBN 1-56881-033-4.
2. T. Bylander, D. Allemang, M.C. M.C. Tanner, and J.R. Josephson. The Computational Complexity of Abduction. *Artificial Intelligence*, 49:25–60, 1991.
3. D.J. Clancy and B.K. Kuipers. Model Decomposition and Simulation: A component based qualitative simulation algorithm. In *AAAI-97*, 1997.
4. J. DeKleer. An Assumption-Based TMS. *Artificial Intelligence*, 28:163–196, 1986.
5. J. Doyle. A Truth Maintenance System. *Artificial Intelligence*, 12:231–272, 1979.
6. S. Easterbrook. *Elicitation of Requirements from Multiple Perspectives.* PhD thesis, Imperial College of Science Technology and Medicine, University of London, 1991. Available from `http://research.ivv.nasa.gov/~steve/papers/index.html`.

7. S. Easterbrook. Handling conflicts between domain descriptions with computer-supported negotiation. *Knowledge Acquisition*, 3:255–289, 1991.
8. B. Feldman, P. Compton, and G. Smythe. Hypothesis Testing: an Appropriate Task for Knowledge-Based Systems. In *4th AAAI-Sponsored Knowledge Acquisition for Knowledge-based Systems Workshop Banff, Canada*, 1989.
9. A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibe. Inconsistency Handling In Multi-Perspective Specification. *IEEE Transactions on Software Engineering*, 20(8):569–578, 1994.
10. H.N. Gabow, S.N. Maheshwari, and L. Osterweil. On Two Problems in the Generation of Program Test Paths. *IEEE Trans. Software Engrg*, SE-2:227–231, 1976.
11. A. Hunter and B. Nuseibeh. Analysing Inconsistent Specifications. In *International Symposium on Requirements Engineering*, pages 78–86, 1997.
12. A.C. Kakas, R.A. Kowalski, and F. Toni. The Role of Abduction in Logic Programming. In C.J. Hogger D.M. Gabbay and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming 5*, pages 235–324. Oxford University Press, 1998.
13. B. Kuipers. Qualitative Simulation. *Artificial Intelligence*, 29:229–338, 1986.
14. T.J. Menzies. *Principles for Generalised Testing of Knowledge Bases.* PhD thesis, University of New South Wales. Avaliable from `http://www.cse.unsw.edu.au/~timm/pub/docs/95thesis.ps.gz`, 1995.
15. T.J. Menzies. On the Practicality of Abductive Validation. In *ECAI '96*, 1996. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/96abvalid.ps.gz`.
16. T.J. Menzies. Applications of Abduction: Knowledge Level Modeling. *International Journal of Human Computer Studies*, 45:305–355, September, 1996. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/96abkl1.ps.gz`.
17. T.J. Menzies and P. Compton. Applications of Abduction: Hypothesis Testing of Neuroendocrinological Qualitative Compartmental Models. *Artificial Intelligence in Medicine*, 10:145–175, 1997. Available from `http://www.cse.unsw.edu.au/~timm/pub/docs/96aim.ps.gz`.
18. B. Nuseibeh. To Be *and* Not to Be: On Managing Inconsistency in Software Development. In *Proceedings of 8th International Workshop on Software Specification and Design (IWSSD-8)*, pages 164–169. IEEE CS Press., 1997.
19. P. O'Rourke. Working Notes of the 1990 Spring Symposium on Automated Abduction. Technical Report 90-32, University of California, Irvine, CA., 1990. September 27, 1990.
20. D. Plexousakis. Semantical and Ontological Considerations in Telos: a Language for Knowledge Representation. *Computational Intelligence*, 9(1), February 1993.
21. A.D. Preece. Principles and Practice in Verifying Rule-based Systems. *The Knowledge Engineering Review*, 7:115–141, 2 1992.
22. R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, 13:81–132, 1980.
23. C. Rich and Y.A. Feldman. Seven Layers of Knowledge Represeentation and Reasoning in Support of Software Development. *IEEE Transactions on Software Engineering*, 18(6):451–469, June 1992.
24. B. Selman and H.J. Levesque. Abductive and Default Reasoning: a Computational Core. In *AAAI '90*, pages 343–348, 1990.

This article was processed using the LaTeX macro package with LLNCS style