Interactive debugging of knowledge-bases
(extended abstract)

Gerhard Friedrich
Institut für Angewandte Informatik
Alpen-Adria Universität Klagenfurt, Austria
e-mail: gerhard.friedrich@aau.at

1 Introduction
The identification and repair of faults in descriptions is a highly complex and expensive task. Descriptions can be formulated for many different purposes such as specifying configurations, hardware designs, algorithms, and requirements of systems. In order to develop technical systems by employing description languages, engineering departments require efficient processes for fault avoidance, testing, and debugging. In particular, the main diagnosis problem in the Engineering Center of Electronics at Siemens AG in order to provide high quality electronic systems has been the localization of faults in software programs, requirements, specifications, and design descriptions. This need for supporting the engineering of hardware and software systems increased over the last decades, since the volume and complexity of implemented functions are sharply increasing and additional formal descriptions for supporting the design processes are being employed.

Moreover the application of declarative descriptions in various technical and non-technical areas (e.g. biomedical domains) is surging and requires new forms of fault localization techniques. Traditional debugging methods which are based on the control flow are not applicable since the fundamental idea of declarative descriptions is to avoid the formulation of a control flow.

First-principles diagnosis [1; 2; 3] (also termed model-based diagnosis (MBD) [4]) is a general approach for diagnosing systems and therefore can be applied not only to hardware systems but also to various forms of descriptions. Consequently, MBD offers a solution to localize faults in various kinds of logic based descriptions such as constraints, answer set programming, temporal logical specifications, and description logic.

For the engineering of software systems tests play a crucial role. The test-driven development approach is based on an iterative cycle of formulating tests, discovering failures, and re-factoring the software. Our goal is to support this development approach also for knowledge engineering. However, for a given set of tests presenting the diagnoses (or just the minimal cardinality diagnoses) is not informative because of the huge number of possibilities to change a knowledge-base. Therefore, we extend static diagnosis to interactive diagnosis (as introduced by [13]) in order to localize precisely the faults in descriptions.

In the following, we will briefly review the main contributions for the diagnosis of various descriptions and provide a summary of the latest developments of applying first-principle diagnosis to knowledge-bases which are expressed by a set of logical sentences.

2 Related work
One of the first application of first-principles diagnosis to descriptions was the debugging of PROLOG programs [5]. This work laid the foundation for the diagnosis of hardware descriptions based on VHDL [6]. During the engineering of hardware, the design descriptions are iteratively refined. The sources of contradictions between these refinements can be located by model-based diagnosis methods. These ideas were adopted for the debugging of procedural programs [7] and for the diagnosis of faulty configuration knowledge-bases [8].

Model-based diagnosis is now applied in many areas dealing with the localization of errors in descriptions, i.e.

- Software [9; 10; 11]
- Software-extended systems [12]
- Plans [13; 14]
- Web services [15; 16; 17]
- Hardware designs [18]
- Recommender systems [19; 20; 21]
- Spreadsheets [22; 23]
- Knowledge-bases expressed by constraints, description logic, and logic programming [24; 25; 26; 27; 28; 29; 30]
- Requirements [31]
- Specifications [32; 33]
- Believes for robot programs [34]

As it turned out the method of first-principles diagnosis is indeed applicable to many diagnosis problems which all have in common the search for a set of failure and health assumptions such that specified conditions (like consistency) are fulfilled.

3 Interactive diagnosis of knowledge-bases
Interactive diagnosis of knowledge-bases provides tools for implementing a test-driven development approach. In [25] we introduce a general diagnosis method which allows the specification of positive and negative test cases. These test cases are exploited to characterize valid knowledge-bases. In particular, a valid knowledge-base $KB^*$ has to be consistent and moreover all positive (negative) test cases must
(not) be implied by $KB^*$. In case of a non-valid knowledge-base $KB$ our method provides minimal sets of sentences of $KB$ which must be changed s.t. a valid knowledge-base can be formulated. The implementation assumes a monotonic semantics of the knowledge-representation and can employ any suitable theorem prover for generating diagnoses and minimal conflicts.

However, the huge number of diagnoses may not allow the knowledge engineer to decide which parts of $KB$ have to be exchanged. Therefore, we provide a method [27] that generates queries for locating the minimal set of sentences which must be changed in order to satisfy the formulated requirements. Our interactive diagnosis method iteratively computes a set of diagnoses and generates a query consisting of a set of facts. The query is constructed in such a way that providing an answer results in the elimination of diagnoses. The knowledge engineer has to decide whether or not this set of facts must be entailed by a valid knowledge base. In case the knowledge engineer cannot answer this question a new query is generated. The iterative process stops if eventually the set of most probable diagnoses (or preferred diagnoses using cardinality as measure) contains one diagnosis which is significantly more likely than all others.

In order to minimize the number of queries, the utility of queries can be estimated by the expected entropy scoring function which exploits prior fault information, i.e. the probability that a sentence is incorrectly formulated. However, in practical domains we might not have reliable information about the likelihood of faults. The split-in-half query-generation strategy does not require fault probabilities and therefore can be applied if no prior fault information is available. This strategy generates queries such that half of the diagnoses are eliminated after the query is answered. However, query-generation strategies such as the minimization of the expected entropy outperform the split-in-half approach if reliable prior fault information is available. We developed a method [28] that combines both strategies by reinforcement learning. The method generates queries by a weighted combination of the entropy and split-in-half strategy and measures the success of the query in eliminating diagnoses. If the prior faulty probabilities are successfully exploited for eliminating diagnoses then the entropy approach is favored otherwise the split-in-half strategy gains more weight.

Finally, we encountered in our evaluations cases with high cardinality faults. These cases are particularly challenging for general diagnosis methods which try to generate a set of most probable diagnoses. We therefore dropped the requirement to generate the set of most probable diagnoses for query-generation but compute a set of diagnoses based on the direct generation method [29]. Thus the computation of conflicts is avoided. By this method [35] we were able to identify the target diagnosis (i.e. the diagnosis which is used to repair a knowledge-base) for cases where traditional diagnosis methods fail. I.e. those methods which compute the set of minimum cardinality diagnoses based on conflicts and minimal hitting set generation.

All the described interactive diagnosis methods are motivated and evaluated by numerous ontologies comprising thousands of axioms. These evaluations show the applicability of model-based diagnosis for debugging large sized knowledge-bases.

4 Conclusions

The diagnosis of descriptions is a large application area of model-based diagnosis. In particular, the engineering of descriptions requires efficient debugging tools. In case the descriptions are declarative, model-based methods provide a solution. We have designed various methods for supporting a test-driven development approach for knowledgebases which are formulated by a set of logical sentences. In order to find a target diagnosis queries are generated and their number is minimized. We also have provided methods which are able to deal with application scenarios where the prior fault information is not reliable or high cardinality faults are present.

Nevertheless, depending on the employed logic and the size of the knowledge-base the diagnosis process is computationally expensive. Consequently, the development of further heuristics and strategies to deal with this computational complexity has high priority on the research agenda.

References


[33] Ingo Pill and Thomas Quaritsch. Behavioral diagnosis of IIL specifications at operator level. In Rossi [37].


