Conflict Management in Interactive Financial Service Selection

Alexander Felfernig[1] and **Martin Stettinger**[1]

Abstract. Knowledge-based systems are often used to support search and navigation in a set of financial services. In a typical process users are defining their requirements and the system selects and ranks alternatives that seem to be appropriate. In such scenarios situations can occur in which requirements can not be fulfilled and alternatives (repairs) must be proposed to the user. In this paper we provide an overview of model-based diagnosis techniques that can be applied to indicate ways out from such a "no solution could be found" dilemma. In this context we focus on scenarios from the domain of financial services.

1 Introduction

Knowledge-based systems such as recommenders [2, 18] and configurators [6, 9, 28] are often used to support users (customers) who are searching for solutions fitting their wishes and needs. These systems select and also rank alternatives of relevance for the user. Examples of such applications are knowledge based recommenders that support users in the identification of relevant financial services [10, 11] and configurators that actively support service configuration [12, 20].

The mentioned systems have the potential to improve the underlying business processes, for example, by reducing error rates in the context of order recording and by reducing time efforts related to customer advisory. Furthermore, customer domain knowledge can be improved by recommendation and configuration technologies; through the interaction with these systems customers gain a deeper understanding of the product domain and – as a direct consequence – less efforts are triggered that are related to the explanation of basic domain aspects. For a detailed overview of the advantages of applying such technologies we refer the reader to [9].

When interacting with knowledge-based systems, situations can occur where no recommendation or configuration can be identified. In order to avoid inefficient manual adaptations of requirements, techniques can be applied which automatically determine repair actions that allow to recover from an inconsistency. For example, if a customer is interested in financial services with high return rates but at the same time does not accept risks related to investments, no corresponding solution will be identified.

There are quite different approaches to deal with the so-called *no solution could be found dilemma* – see Table 1. In the context of this paper we will focus on the application of the concepts of model-based diagnosis [27, 5]. A first application of model-based diagnosis to the automated identification of erroneous constraints in knowledge bases is reported in Bakker et al. [1]. In their work the authors show how to model the task of identifying faulty constraints in a knowledge base as a diagnosis task. Felfernig et al. [8] extend the approach of Bakker et al. [1] by introducing concepts that allow the automated debugging of (configuration) knowledge bases on the basis of test cases. If one or more test cases fail within the scope of regression testing, a diagnosis process is activated that determines a minimal set of constraints in such a way that the deletion of these constraints guarantees that each test case is consistent with the knowledge base. Model-based diagnosis [27] relies on the existence of conflict sets which represent minimal sets of inconsistent constraints. Conflict sets can be determined by conflict detection algorithms such as QUICKXPLAIN [19].

Beside the automated testing and debugging of inconsistent knowledge bases, model-based diagnosis is also applied in situations where the knowledge base per se is consistent but a set of customer requirements induces an inconsistency. Felfernig et al. [8] also sketch an approach to the application of model-based diagnosis to the identification of minimal sets of fault requirements. Their approach is based on breadth-first search that uses diagnosis cardinality as the only ranking criteria.

A couple of different approaches to the determination of personalized diagnoses for inconsistent requirements have been proposed. DeKleer [4] introduces concepts for the probability-based identification of leading diagnoses. O'Sullivan et al. [25] introduce the concept of representative explanations (diagnosis sets) where each existing diagnosis element is contained in at least one diagnosis of a representative set of diagnoses. Felfernig et al. [13] show how to integrate basic recommendation algorithms into diagnosis search and with this to increase the prediction quality (in terms of precision) of diagnostic approaches. Felfernig et al. [14] extend this work and compare different personalization approaches with regard to their prediction quality and the basis of real-world datasets. Based on the concepts of QUICKXPLAIN, Felfernig et al. [15] introduced FASTDIAG which improves the efficiency of diagnosis search by omitting the calcualation of conflicts as a basis for diagnosis calculation. This diagnostic approach is also denoted as direct diagnosis [17]. The applicability of FASTDIAG has also been shown in SAT solving scenarios [23].

Different types of knowledge-based systems have already been applied to support the interactive selection and configuration of fi-

[1] Applied Software Engineering, Institute for Software Technology, Graz University of Technology, Austria, email: {felfernig, stettinger}@ist.tugraz.at.

Topic	Reference
Foundations of model-based diagnosis	Reiter 1987 [27], DeKleer et al. 1992 [5]
Conflict detection and model-based diagnosis of inconsistent constraint satisfaction problems (CSPs)	Bakker et al. 1993 [1]
Regression testing and automated debugging of configuration knowledge bases using model-based diagnosis (breadth-first search)	Felfernig et al. 2004 [8]
Identification of minimal diagnoses for user requirements for the purpose of consistency preservation (breadth-first search)	
Identification of preferred minimal conflict sets on the basis of a divide-and-conquer based algorithm (QUICKXPLAIN)	Junker 2004 [19]
Identification of representative explanations (each existing diagnosis element is contained in at least one diagnosis of the result set)	O'Sullivan et al. 2007 [25]
Identification of personalized diagnoses on the basis of recommendation algorithms	Felfernig et al. 2009,2013 [13, 14]
Probability based identification of leading diagnoses	DeKleer 1990 [4]
Identification of preferred minimal diagnoses on the basis of a divide-and-conquer based algorithm (FASTDIAG)	Felfernig et al. 2012 [15]
Preferred minimal diagnoses for SAT based knowledge representations	Marques-Silva et al. 2013 [23]

Table 1. Overview of research related to conflict management in knowledge-based systems.

nancial services. Fano and Kurth [7] introduce an approach to the visualization and planning of financial service portfolios. The simulation is based on an integrated model of a human's household and interdependencies between different financial decisions. Felfernig et al. [10, 11] show how to apply knowledge-based recommender applications for supporting sales representatives in their dialogs with customers. Major improvements that can be expected from such an approach are less errors in the offer phase and more time for additional customer meetings. An approach to apply the concepts of cased-based reasoning [21] for the purpose of recommending financial services is introduced by Musto et al. [24].

The major focus of this paper is to provide an overview of techniques that help to recover from inconsistent situations in an automated fashion. In this context we show how inconsistencies can be identified and resolved. The major contributions of this paper are the following: (1) we provide an overview of error identification and repair techniques in the context of financial services recommendation and configuration. (2) We show how diagnosis and repair techniques can be applied on the basis of different knowledge representations (CSPs as well as table-based representations). (3) We provide an outlook of major issues for future work.

The remainder of this paper is organized as follows. In Section 2 we introduce basic definitions of a constraint satisfaction problem (CSP) and a corresponding solution. On the basis of these definitions we introduce a first working example from the financial services domain. Thereafter (in Section 3) we introduce a basic definition of a diagnosis task and show how diagnoses and repairs for inconsistent user requirements can be determined. In Section 4 we switch from constraint-based to table-based knowledge representations where (personalized) solutions are determined on the basis of conjunctive queries [13]. In Section 5 we provide one further example of consistency management in the loan domain. In Section 6 we discuss issues for future work. With Section 7 we conclude the paper.

2 Constraint-based Representations

Constraint Satisfaction Problems (CSPs) [16, 22] are successfully applied in many industrial scenarios such as scheduling [26], configuration [9], and recommender systems [18]. The popularity of this type of knowledge representation can be explained by the small set of representation concepts (only variables, related domains, and constraints have to be defined) and the still high degree of expressivity.

Definition 1 (Constraint Satisfaction Problem (CSP) and Solution). A constraint satisfaction problem (CSP) can be defined as a triple (V, D, C) where $V = \{v_1, v_2, ..., v_n\}$ represents a set of variables, $dom(v_1), dom(v_2), ..., dom(v_n)$ represents the corresponding variable domains, and $C = \{c_1, c_2, ..., c_m\}$ represents a set of constraints that refer to corresponding variables and reduce the number of potential solutions. A solution for a CSP is defined by an assignment A of all variables in V where A is consistent with the constraints in C.

Usually, user requirements are interpreted as constraints $CREQ = \{r_1, r_2, ..., r_q\}$ where r_i represent individual user requirements. In this paper we assume that the constraints in C are consistent and inconsistencies are always induced by the constraints in $CREQ$. If such a situation occurs, we are interested in the elements of $CREQ$ which are responsible for the given inconsistency. On the basis of a first example we will now provide an overview of diagnosis techniques that can be used to recover from such inconsistent situations. An example of a CSP in the domain of financial services is the following. For simplicity we assume that each variable has the domain {*low, medium, high*}.

- $V = \{av, wr, rr\}$
- $dom(av) = dom(wr) = dom(rr) = \{low, medium, high\}$
- $C = \{c_1 : \neg(av = high \wedge wr = high), c_2 : \neg(wr = low \wedge rr = high), c_3 : \neg(rr = high \wedge av = high)\}$

An overview of the variables of this CSP is given in Table 2.

variable	description	$r_i \in CREQ$
av	availability	$r_1 : av = high$
wr	willingness to take risks	$r_2 : wr = low$
rr	expected return rate	$r_3 : rr = high$

Table 2. Overview of variables used in the example CSP definition.

In addition to this basic CSP definition we introduce an example set of customer requirements $CREQ = \{r_1 : av = high, r_2 : wr = low, r_3 : rr = high\}$ which is inconsistent with the constraints defined in C. On the basis of this simplified financial service knowledge base defined as a CSP we will now show how inconsistencies induced by customer requirements can be identified and resolved.

3 Diagnosis & Repair of Inconsistent Constraints

In our working example, the requirements $CREQ$ and the set of constraints C are inconsistent, i.e., inconsistent($CREQ \cup C$). In such situations we are interested in a minimal set of requirements that have to be deleted or adapted such that consistency is restored. Consistency resolution is in many cases based on the resolution of conflicts. In our case, a minimal conflict is represented by a minimal set of requirements in $CREQ$ that have to be deleted or adapted such that consistency can be restored.

Definition 2 (Conflict Set). A conflict set CS is a subset of $CREQ$ s.t. inconsistent ($CS \cup C$). A conflict set is minimal if there does not exist another conflict set CS' with $CS' \subset CS$. A minimal cardinality conflict set CS is a minimal conflict set with the additional property that there does not exist another minimal conflict CS' with $|CS'| < |CS|$.

Minimal conflict sets can be determined on the basis of conflict detection algorithms such as QUICKXPLAIN [19]. They can be used to derive diagnoses. In our case, a diagnosis Δ represents a set of requirements that have to be deleted from $CREQ$ such that $C \cup (CREQ - \Delta)$ is consistent, i.e., diagnoses help to restore the consistency between $CREQ$ and C.

Definition 3 (Diagnosis Task and Diagnosis). A diagnosis task can be defined as a tuple ($C, CREQ$) where C represents a set of constraints in the knowledge base and $CREQ$ represents a set of customer requirements. Δ is a diagnosis if $CREQ - \Delta \cup C$ is consistent. A diagnosis Δ is minimal if there does not exist a diagnosis Δ' with $\Delta' \subset \Delta$. Furthermore, Δ is a minimal cardinality diagnosis if there does not exist a diagnosis Δ' with $|\Delta'| < |\Delta|$.

A standard approach to the determination of diagnoses is based on the construction of a hitting set directed acyclic graph (HSDAG) [27] where minimal conflict sets are successively resolved in the process of HSDAG construction (an example is depicted in Figure 1). In the context of our example of C and $CREQ$, a first minimal conflict set that could be returned by an algorithm such as QUICKXPLAIN [19] is $CS_1 : \{r_1, r_3\}$.

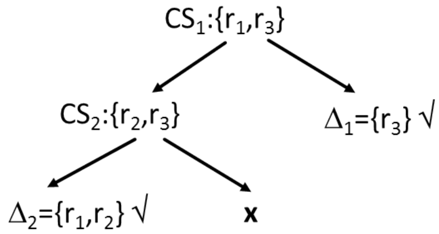

Figure 1. Hitting Set Directed Acyclic Graph (HSDAG) for requirements $CREQ = \{r_1 : av = high, r_2 : wr = low, r_3 : rr = high\}$.

There are two possibilities of resolving CS_1, either by deleting requirement r_1 or by deleting requirement r_3. If we delete r_3 (see Figure 1), we managed to identify the first minimal diagnosis $\Delta_1 = \{r_3\}$ which is also a minimal cardinality diagnosis. The second option to resolve CS_1 is to delete r_1. In this situation, another conflict exists in $CREQ$, i.e., a conflict detection algorithm would return $CS_2 : \{r_2, r_3\}$. Again, there are two possibilities to resolve the conflict (either by deleting r_2 or by deleting r_3). Deleting r_3 leads to a diagnosis which is not minimal since $\{r_3\}$ itself is already a diagnosis. Deleting r_2 leads to the second minimal diagnosis which is $\Delta_2 = \{r_1, r_2\}$.

The diagnoses Δ_1 and Δ_2 are indicators of minimal changes that need to be performed on the existing set of requirements such that a consistency between $CREQ$ and C can be restored. The issue of finding concrete repair actions for the requirements contained in a diagnosis will be discussed later in this paper.

There can be quite many alternative diagnoses. In this context it is not always clear which diagnosis should be selected or in which order alternative diagnoses should be shown to the user. In the following we present one approach to rank diagnoses. The approach we sketch is based on multi-attribute utility theory [29] where we assume that customers provide weights for each individual requirement. In the example depicted in Table 3, two customers specified their preferences in terms of weights for each requirement. For example, customer 1 specified a weight of 0.7 for the requirement $r_3 : rr = high$, i.e., the attribute rr is of highest importance for the customer. These weights can be exploited for ranking a set of diagnoses.

Formula 1 can be used for determining the overall importance (imp) of a set of requirements (RS). The higher the importance the lower the probability that these requirements are element of a diagnosis shown to the customer. Requirement r_3 has a high importance for customer 1, consequently, the probability that r_3 is contained in a diagnosis shown to customer 1 is low.

$$imp(RS) = importance(RS) = \Sigma_{r \in RS} weight(r) \quad (1)$$

Formula 2 can be used to determine the relevance of a partial or complete (minimal) diagnosis, i.e., this formula can be used to rank

customer	weight($r_1 : av = high$)	weight($r_2 : wr = low$)	weight($r_3 : rr = high$)
1	0.1	0.2	0.7
2	0.3	0.5	0.2

Table 3. Individual weights regarding the importance of the requirements $CREQ = \{r_1, r_2, r_3\}$.

diagnoses with regard to their relevance for the customer. The higher the relevance of a diagnosis, the higher the ranking of the diagnosis in a list of diagnoses shown to the customer.

$$rel(\Delta) = relevance(\Delta) = \frac{1}{importance(\Delta)} \quad (2)$$

Tables 4 and 5 show the results of applying Formulae 1 and 2 to the customer preferences (weights) shown in Table 3. For customer 1 (see Table 4), diagnosis $\Delta_2 = \{r_1, r_2\}$ has the highest relevance. For customer 2 (see Table 5), diagnosis $\Delta_1 = \{r_3\}$ has the highest relevance. Consequently, diagnosis Δ_2 is the first one that will be shown to customer 1 and diagnosis Δ_1 is the first one that will be shown to customer 2.

diagnosis Δ_j	importance(Δ_j)	relevance(Δ_j)
$\Delta_1 : \{r_3\}$	0.7	1.43
$\Delta_2 : \{r_1, r_2\}$	0.3	3.33

Table 4. Diagnosis with highest relevance (*rel*) determined for *customer 1*: $\Delta_2 = \{r_1, r_2\}$.

diagnosis Δ_j	importance(Δ_j)	relevance(Δ_j)
$\Delta_1 : \{r_3\}$	0.2	5.0
$\Delta_2 : \{r_1, r_2\}$	0.8	1.25

Table 5. Diagnosis with highest relevance (*rel*) determined for *customer 2*: $\Delta_1 = \{r_3\}$.

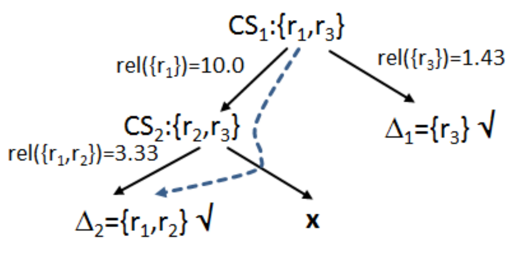

Figure 2. Personalized diagnosis determined for $CREQ$ and the individual importance weights defined in Table 3 (for customer1). In this example, Δ_2 is the preferred diagnosis since $relevance(\Delta_2) > relevance(\Delta_1)$.

On the basis of the relevance values depicted in Table 4, Figure 2 depicts a HSDAG [27] with additional annotations regarding diagnosis relevance (rel). The higher the relevance of a (partial) diagnosis, the higher the ranking of the corresponding diagnosis.

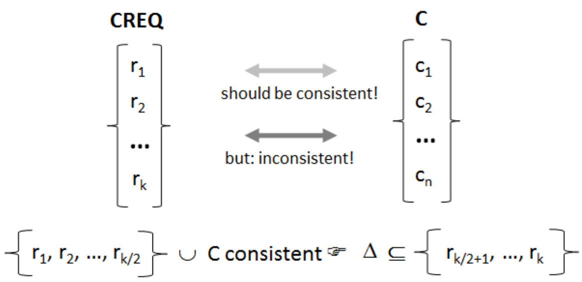

Figure 3. FASTDIAG approach to diagnosis determination. $CREQ$ represents a set of customer requirements and C represents a set of constraints. The algorithm is based on a divide-and-conquer approach: if $\{r_1, r_2, ..., r_{k/2}\}$ is consistent with C then diagnosis search can be continued in $\{r_{k/2+1}...r_k\}$. Δ is a diagnosis if $CREQ - \Delta \cup C$ is consistent.

The afore discussed approaches to diagnosis determination are based on the construction of a HSDAG [27]. Due to the fact that conflicts have to determined explicitly when following this approach, diagnosis determination does not scale well [13, 14]. The FASTDIAG algorithm [15] tackles this challenge by determining minimal and preferred diagnoses without the need of conflict detection. This algorithm has shown to have the same predictive quality as HSDAG based algorithms that determine diagnoses in a breadth-first search regime. The major advantage of FASTDIAG is a high-performance diagnosis search for the leading diagnoses (first-n diagnoses).

FASTDIAG is based on the principle of *divide and conquer* – see Figure 3: if a set of requirements $CREQ$ is inconsistent with a corresponding set of constraints C and the first part $\{r_1, r_2, ..., r_{k/2}\}$ of $CREQ$ is consistent with C then diagnosis search can focus on $\{r_{k/2+1}, ..., r_k\}$, i.e., can omit the requirements in $\{r_1, r_2, ..., r_{k/2}\}$. A detailed discussion of FASTDIAG can be found in [15].

Determination of Repair Actions. Repair actions for diagnosis elements can be interpreted as changes to the originial set of requirements in $CREQ$ in such a way that at least one solution can be identified. If we assume that $CREQ$ is a set of unary constraints that are inconsistent with C and Δ is a corresponding diagnosis, then a set of repair actions $R = \{a_1, a_2, ..., a_l\}$ can be identified by the consistency check $CREQ - \Delta \cup C$ where a_j (a variable assignment) is a repair for the constraint r_j if r_j is in Δ.

In this section we took a look at different approaches that support the determination of diagnoses in situations where a given set of requirements becomes inconsistent with the constraints in C. In the following we will take a look at an alternative knowledge representation where tables (instead of CSPs) are used to represent knowledge

id	return rate p.a. (rr)	runtime in yrs. (rt)	risk level (wtr)	shares percentage (sp)	acessibility (acc)	bluechip(bc)
1	4.2	3.0	A	0.0	no	yes
2	4.7	3.7	B	10.0	yes	yes
3	4.8	3.5	A	10.0	yes	yes
4	5.2	4.0	B	20.0	yes	no
5	4.3	3.5	A	0.0	yes	yes
6	5.6	5.0	C	30.0	no	no
7	6.7	6.0	C	50.0	yes	no
8	7.9	7.0	C	50.0	no	no

Table 6. Investment products: *return rate p.a. (rr)*, *runtime in years (rt)*, *risk level (wtr)*, *shares percentage (sp)*, *accessibility (acc)*, and *bluechip (bc)*.

customer	weight($r_1 : rr \geq 5.5$)	weight($r_2 : rt = 3.0$)	weight($r_3 : acc = yes$)	weight($r_4 : bc = yes$)
1	0.7	0.1	0.1	0.1
2	0.1	0.7	0.1	0.1

Table 7. Individual weights regarding the importance of the requirements $CREQ$ =$\{r_1, r_2, r_3, r_4\}$.

about financial services. Again, we will show how to deal with inconsistent situations.

4 Table-based Representations

In Section 3 we analyzed different ways of diagnosing inconsistent CSPs [16, 22]. We now show how diagnosis can be performed on a predefined set of solutions, i.e., a table-based representation. Table 6 includes an example set of investment products. The set of financial services $\{1, 2, ..., 8\}$ is stored in an item table T [13] – T can be interpreted as an explicit enumeration of the possible solutions (defined by the set C in Section 2). Furthermore, we assume that the customer has specified a set of requirements $CREQ$ = $\{r_1 : rr \geq 5.5, r_2 : rt = 3.0, r_3 : acc = yes, r_4 : bc = yes\}$. The existence of a financial service in T that is able to fulfill all requirements can be checked by a relational query $\sigma_{[CREQ]}T$ where $CREQ$ represents a set of selection criteria and T represents the corresponding product table.

An example query on the product table T could be $\sigma_{[rr \geq 5.5]}T$ which would return the financial services {6,7,8}. For the query $\sigma_{[r_1,r_2,r_3,r_4]}T$ there does not exist a solution. In such situations we are interested in finding diagnoses that indicate minimal sets of requirements in $CREQ$ that have to be deleted or adapted in order to be able to identify a solution.

Definition 4 (Conflict Sets in Table-based Representations). A conflict set CS is a subset of $CREQ$ s.t. $\sigma_{[CS]}T$ returns an empty result set. Minimality properties of conflict sets are the same as introduced in Definition 2.

A diagnosis task and a corresponding diagnosis in the context of table-based representations can be defined as follows.

Definition 5 (Diagnosis in Table-based Representations). A diagnosis task can be defined as a tuple $(T, CREQ)$ where T represents a product table and $CREQ$ represents a set of customer requirements. Δ is a diagnosis if $\sigma_{[CREQ-\Delta]}T$ returns at least one solution. Minimality properties of diagnoses are the same as in Definition 3.

The requirements $r_j \in CREQ$ are inconsistent with the items included in T (see Table 6), i.e., there does not exist a financial service in T that completely fulfills the user requirements in $CREQ$. *Minimal conflict sets* that can be derived for $CREQ$ = $\{r_1 : rr \geq 5.5, r_2 : rt = 3.0, r_3 : acc = yes, r_4 : bc = yes\}$ are $CS_1 : \{r_1, r_2\}$, $CS_2 : \{r_2, r_3\}$, and $CS_3 : \{r_1, r_4\}$. The determination of the corresponding diagnoses is depicted in Figure 4.

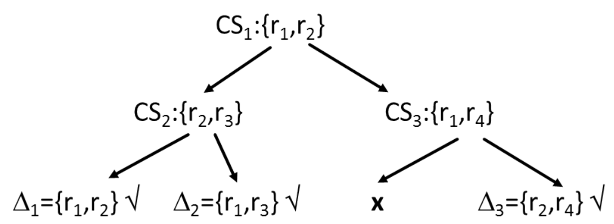

Figure 4. Hitting Set Directed Acyclic Graph (HSDAG) for requirements $CREQ$ = $\{r_1 : rr \geq 5.5, r_2 : rt = 3.0, r_3 : acc = yes, r_4 : bc = yes\}$.

Diagnoses are determined in the same fashion as discussed in Section 2. Minimal diagnoses that can be derived from the conflict sets CS_1, CS_2, and CS_3 are $\Delta_1 : \{r_1, r_2\}$, $\Delta_2 : \{r_1, r_3\}$ and $\Delta_3 : \{r_2, r_4\}$ (see Figure 4).

Again, the question arises which of the diagnoses has the highest relevance for the user (customer). Table 7 depicts the importance distributions for the requirements of our example. Based on the importance distributions depicted in Table 7 we can derive a preferred diagnosis (see Figure 5). Diagnosis Δ_3 will be first shown to customer 1 since Δ_3 has the highest evaluation in terms of relevance (see Formula 2). The first diagnosis shown to customer 2 is Δ_2.

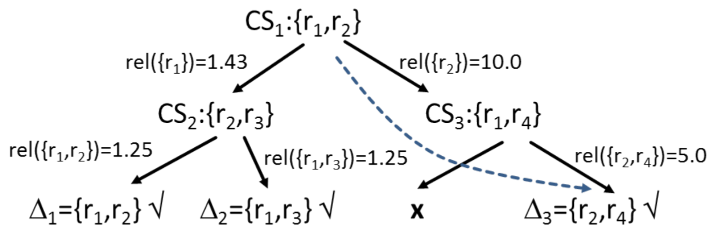

Figure 5. Personalized diagnoses determined for $CREQ$ and the individual importance weights defined in Table 7 (for customer 1). In this example, Δ_3 is the preferred diagnosis.

diagnosis Δ_j	importance(Δ_j)	relevance(Δ_j)
$\Delta_1 : \{r_1, r_2\}$	0.8	1.25
$\Delta_2 : \{r_1, r_3\}$	0.8	1.25
$\Delta_3 : \{r_2, r_4\}$	0.2	5.0

Table 8. Diagnosis with highest relevance (*rel*) determined for *customer 1*: $\Delta_3 = \{r_2, r_4\}$.

diagnosis Δ_j	importance(Δ_j)	relevance(Δ_j)
$\Delta_1 : \{r_1, r_2\}$	0.8	1.25
$\Delta_2 : \{r_1, r_3\}$	0.2	5.0
$\Delta_3 : \{r_2, r_4\}$	0.8	1.25

Table 9. Diagnosis with highest relevance (*rel*) determined for *customer 2*: $\Delta_2 = \{r_1, r_3\}$.

id	creditworthiness(cw)	loan limit(ll)	runtime in yrs.(rt)	interest rate (ir)
1	1	30.000	5.0	3%
2	2	25.000	5.0	4%
3	3	20.000	5.0	5%
4	1	40.000	6.0	4%
5	2	35.000	6.0	5%
6	3	30.000	7.0	5.2%
7	1	40.000	5.0	3%
8	2	35.000	5.0	3.5%
9	3	30.000	5.0	5%

Table 10. Loans: *creditworthiness (cw)*, *loan limit (ll)*, *runtime in years (rt)*, and *interest rate (ir)*.

5 An Additional Example: Selection of Loans

As a third example we introduce the domain of loans. The entries in Table 10 represent different loan variants that can be chosen by customers. Customers can specify their requirements on the basis of the variables depicted in Table 11. Furthermore, the different loan variants are characterized by their *expected creditworthiness (cw)*, *loan limit (ll)*, *runtime in yrs. (rt)*, and *interest rate (ir)*. These variables are basic elements of the definition of the following Constraint Satisfaction Problem (CSP).

variable	description	$r_i \in CREQ$
ccw	current creditworthiness	$r_1 : ccw = 3$
ils	intended loan sum	$r_2 : ils = 30.000$
mpp	maximum periodical payment	–
irt	intended runtime	$r_3 : irt = 6yrs.$
pir	preferred interest rate	$r_4 : pir = 4.5\%$

Table 11. Overview of variables used in the example CSP definition (loans).

- V = {ccw, ils, mpp, irt, pir, cw, ll, rt, ir}
- dom(ccw) = dom(cw) = {1,2,3}; dom(ils) = dom(ll) = float; dom(mpp) = float; dom(irt) = dom(rt) = integer; dom(pir) = dom(ir) = integer.
- C = $\{c_1 : ccw \leq cw, c_2 : ils \leq ls, c_3 : irt = rt, c_4 : pir \geq ir, c_5 : \mathit{see\ below}, c_{6,7} : \mathit{see\ below}\}$

Constraint c_5 represents the entities of Table 10 in disjunctive normal form, for example, the first table row can be represented as basic constraint $\{cw = 1 \wedge ll = 30.000 \wedge rt = 5.0 \wedge ir = 3\%\}$. The disjunct of all basic constraints is the disjunctive normal form. Constraints $c_{6,7}$ can be used to avoid situations where the periodical payments for a loan exceed the financial resources of the customer.

$$c_6 : mpp \geq \frac{costs(id) + ils}{rt} \tag{3}$$

$$c_7 : costs(id) = ils \times ir(id) \times \frac{(rt(id) + 1)}{2} \tag{4}$$

For the purpose of our example let us assume that the customer has the following requirements: $CREQ = \{r_1 : ccw = 3, r_2 : ils = 30.000, r_3 : irt = 6yrs., r_4 : pir = 4.5\%\}$. Since the customer creditworthiness has been evaluated with 3, only three alternative loan variants are available (the ids 3,6,9). These variants are depicted in Table 12.

id	cw	ll	rt	ir
3	3	20.000	5.0 yrs.	5%
6	3	30.000	7.0 yrs.	5.2%
9	3	30.000	5.0 yrs.	5%

Table 12. Loans accessible for the customer with creditworthiness level 3.

Since $CREQ$ is inconsistent with the constraints in C we could determine minimal diagnoses as indicators for possible adaptations in the requirements. A possible criteria for personalizing diagnosis ranking could be the *costs related to a loan* (see Formula 4). The requirements $CREQ$ include one minimal conflict set which is $CS_1 : \{r_3, r_4\}$. Consequently, there exist two different possibilities to resolve the conflict: one possibility is to change the value for the *intended runtime* (irt) from 6.0 years to 5.0 years and to keep the *preferred interest rate* (pir) as is. The other possibility is to change the preferred interest rate from 4.5% to 6% and to keep the intended runtime as is. The overall loan costs related to these two alternatives are depicted in Table 13. If the overall loan costs are a major criteria then repair alternative 1 would be chosen by the customer, otherwise – if the upper limit for periodical payments is strict – repair alternative 2 will be chosen.

repair alternative	irt	pir	costs	costs per year
1	5.0 yrs.	5.0%	4.500	900.00
2	7.0 yrs.	5.2%	6.240	891.43

Table 13. Loan costs for different repair alternatives.

6 Future Work

A major issue for interactive applications is to guarantee reasonable response times which should be below one second [3]. This goal can not be achieved with standard diagnosis approaches since they typically rely on the (pre-)determination of conflict sets. Although existing divide-and-conquer based diagnosis approaches are significantly faster when determining only leading (preferred) diagnosis, i.e., not all diagnoses have to be determined, there is still a need for improving diagnosis efficiency in more complex settings. In this context, on research issue is the development of so-called anytime diagnosis algorithms that help to determine nearly optimal (e.g., in terms of prediction quality) diagnoses with less computational efforts.

Although the prediction quality of diagnoses significantly increases and numerous recommendation algorithms have already been evaluated, there is still a need for further advancing the state-of-the-art in diagnosis prediction. One research direction is to focus on learning-based approaches that help to figure out which combination of a set of basic diagnosis prediction methods best performs in the considered domain. Such approaches are also denoted as ensemble-based methods which focus on figuring out optimal configurations of basic diagnosis prediction methods.

Efficient calculation and high predictive quality are for sure central issues of future research. Beyond efficiency and prediction quality, intelligent visualization concepts for diagnoses are extremely important. For example, the the context of group decision scenarios where groups of users are in charge of resolving existing inconsistencies in the preferences between group members, visualizations have to be identified that help to restore consistency (consensus) in the group as soon as possible. Such visualizations could focus on visualizing the mental state on individual group members as well visualizing the individual decision behavior (e.g., egoism vs. altruism).

7 Conclusions

In this paper we give an overview of existing approaches to determine diagnoses in situations were no solution can be found. We first provide an overview of existing related work and then focus on basic approaches to determine diagnoses in the context of two knowledge representation formalisms (constraint satisfaction and conjunctive query based approaches). For explanation purposes we introduce three different types of financial services as working examples (basic investment decisions, selection of investment products, and loan selection). On the basis of these examples we sketch the determination of (preferred) diagnoses. Thereafter, we provide a short discussion of open research issues which includes diagnosis efficiency, prediction quality, and intelligent visualization.

REFERENCES

[1] R. Bakker, F. Dikker, F. Tempelman, and P. Wogmim, 'Diagnosing and solving over-determined constraint satisfaction problems', in *IJCAI 1993*, pp. 276–281, Chambery, France, (1993).

[2] R. Burke, A. Felfernig, and M. Goeker, 'Recommender systems: An overview', *AI Magazine*, **32**(3), 13–18, (2011).

[3] S. Card, G. Robertson, and J. Mackinlay, 'The information visualizer, an information workspace', in *Conference on Human Factors in Computing Systems: Reaching Through Technology*, pp. 181–186, New York, NY, USA, (1991).

[4] J. de Kleer, 'Using crude probability estimates to guide diagnosis', *AI Journal*, **45**(3), 381–391, (1990).

[5] J. de Kleer, A. Mackworth, and R. Reiter, 'Characterizing diagnoses and systems', *AI Journal*, **56**(197–222), 57–95, (1992).

[6] A. Falkner, A. Felfernig, and A. Haag, 'Recommendation Technologies for Configurable Products', *AI Magazine*, **32**(3), 99–108, (2011).

[7] A. Fano and S. Kurth, 'Personal Choice Point: Helping Users Visualize What it Means to Buy a BMW', in *International Conference on Intelligent User Interfaces IUI'03*, pp. 46–52, Miami, FL, USA, (2003). ACM, New York, USA.

[8] A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner, 'Consistency-based diagnosis of configuration knowledge bases', *Artificial Intelligence*, **152**(2), 213–234, (2004).

[9] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-based Configuration – From Research to Business Cases*, Elsevier, Morgan Kaufmann, 2014.

[10] A. Felfernig, K. Isak, K. Szabo, and P. Zachar, 'The VITA Financial Services Sales Support Environment', pp. 1692–1699, Vancouver, Canada, (2007).

[11] A. Felfernig and A. Kiener, 'Knowledge-based Interactive Selling of Financial Services with FSAdvisor', in *17th Innovative Applications of Artificial Intelligence Conference (IAAI05)*, pp. 1475–1482, Pittsburgh, Pennsylvania, (2005).

[12] A. Felfernig, J. Mehlau, A. Wimmer, C. Russ, and M. Zanker, 'Konzepte zur flexiblen Konfiguration von Finanzdienstleistungen', *Banking and Information Technology*, **5**(1), 5–19, (2004).

[13] A. Felfernig, M. Schubert, G. Friedrich, M. Mandl, M. Mairitsch, and E. Teppan, 'Plausible repairs for inconsistent requirements', in *21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pp. 791–796, Pasadena, CA, (2009).

[14] A. Felfernig, M. Schubert, and S. Reiterer, 'Personalized diagnosis for over-constrained problems', in *23rd International Conference on Artificial Intelligence (IJCAI 2013)*, pp. 1990–1996, Peking, China.

[15] A. Felfernig, M. Schubert, and C. Zehentner, 'An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets', *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)*, **25**(2), 175–184, (2012).

[16] E. Freuder, 'In pursuit of the holy grail', *Constraints*, **2**(1), 57–61, (1997).

[17] G. Friedrich, 'Interactive Debugging of Knowledge Bases', in *DX'2014*, pp. 1–4, Graz, Austria, (2014).

[18] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems*, Cambridge University Press, 2010.

[19] U. Junker, 'Quickxplain: Preferred explanations and relaxations for over-constrained problems', in *19th National Conference on AI (AAAI04)*, pp. 167–172, San Jose, CA, (2004).

[20] S. Leist and R. Winter, 'Konfiguration von Versicherungsdienstleistungen', *Wirtschaftsinformatik*, **36**(1), 45–56, (1994).

[21] F. Lorenzi and F. Ricci, 'Case-based recommender systems: a unifying view', *Intelligent Techniques for Web Personalization*, 89–113, (2005).

[22] A. Mackworth, 'Consistency in Networks of Relations', *Artificial Intelligence*, **8**(1), 99–118, (1977).

[23] J. Marques-Silva, F. Heras, M. Janota, A. Previti, and A. Belov, 'On computing minimal correction subsets', in *IJCAI 2013*, pp. 615–622, Peking, China, (2013).

[24] C. Musto, G. Semeraro, P. Lops, M. DeGemmis, and G. Lekkas, 'Financial Product Recommendation through Case-based Reasoning and Diversification Techniques', pp. 1–2, Foster City, CA, USA, (2014).

[25] B. O'Sullivan, A. Papadopoulos, B. Faltings, and P. Pu, 'Representative explanations for over-constrained problems', in *AAAI'07*, pp. 323–328, Vancouver, Canada, (2007).

[26] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Springer, 4 edn., 2012.

[27] R. Reiter, 'A theory of diagnosis from first principles', *AI Journal*, **23**(1), 57–95, (1987).

[28] J. Tiihonen and A. Felfernig, 'Towards Recommending Configurable Offerings', *International Journal of Mass Customization*, **3**(4), 389–406, (2010).

[29] D. Winterfeldt and W. Edwards, 'Decision Analysis and Behavioral Research', *Cambridge University Press*, (1986).