FINREC 15

Proceedings of the
1st International Workshop on Personalization & Recommender Systems in Financial Services

APRIL 16, 2015 | GRAZ, AUSTRIA

Edited by Alexander Felfernig, Juha Tiihonen and Paul Blazek

Organized by

TU Graz

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

CYLEDGE

Graz University of Technology
Institute for Software Technology
Inffeldgasse 16b/2
A-8010 Graz
Austria

Alexander Felfernig, Juha Tiihonen, and Paul Blazek, Editors
Proceedings of the 1st International Workshop on Personalization & Recommender Systems in Financial Services
April 16, 2015, Graz, Austria

Chairs

Alexander Felfernig, Graz University of Technology, Austria
Juha Tiihonen, University of Helsinki, Finland
Paul Blazek, cyLEDGE, Austria

Program Committee

Zoran Anišić, University of Novi Sad, Serbia
Mathias Bauer, mineway GmbH, Germany
Shlomo Berkovsky, NICTA, Australia
Paul Blazek, cyLEDGE, Austria
Robin Burke, DePaul University, IL, USA
Kuan-Ta Chen, Academia Sinica, Taiwan
Li Chen, Hong Kong Baptist University, China
Marco De Gemmis, University of Bari, Italy
John O'Donovan, University of California Santa Barbara, CA, USA
Alexander Felfernig, Graz University of Technology, Austria
Gerhard Friedrich, Alpen-Adria-Universitaet Klagenfurt, Austria
Hagen Habicht, CLIC, HHL Leipzig Graduate School of Management, Germany
Dietmar Jannach, TU Dortmund, Germany
Gerhard Leitner, Alpen-Adria-Universitaet Klagenfurt, Austria
Pasquale Lops, University of Bari, Italy
Hans Lundberg, Linnaeus University, Sweden
Eetu Mäkelä, Aalto University, Finland
Birgit Penzenstadler, California State University Long Beach, CA, USA
Giovanni Semeraro, University of Bari, Italy
Ian Sutherland, IEDC-Bled School of Management, Slovenia
Juha Tiihonen, Aalto University, Finland
Nava Tintarev, University of Aberdeen, UK
Shuang-Hong Yang, Twitter Inc., CA, US
Markus Zanker, Alpen-Adria-Universitaet Klagenfurt, Austria

Organizational Support

Martin Stettinger, Graz University of Technology, Austria

Preface

Personalization and recommendation technologies provide the basis for applications that are tailored to the needs of individual users. These technologies play an increasingly important role for financial service providers. The selection of papers of this year’s workshop demonstrates the wide range of techniques including contributions on knowledge-based recommender systems, case-based reasoning, knowledge interchange, psychological aspects of recommender systems in financial services, MediaWiki-based recommendation technologies, smart data analysis and big data, and campaign customization.

The workshop is of interest for both, researchers working in the various fields of personalization and recommender systems as well as for industry representatives. It provides a forum for the exchange of ideas, evaluations, and experiences. As such, this year's workshop on “Personalization & Recommender Systems in Financial Services” aims at providing a stimulating environment for knowledge-exchange among academia and industry and thus building a solid basis for further developments in the field.

Alexander Felfernig, Juha Tiihonen, and Paul Blazek

Contents

Smart Data Analysis for Financial Services

Mathias Bauer[1]

Abstract. This talk addresses opportunities for the application of intelligent data analysis techniques at various stages of the value added chain for financial services. After introducing some basic notions and explaining the fundamental steps of data mining, we will have a closer look at various recent and ongoing projects and discuss issues of practical relevance such as data quality and expert knowledge. The talk concludes with some remarks on the potential impact of new developments, e. g. in the context of Big Data.

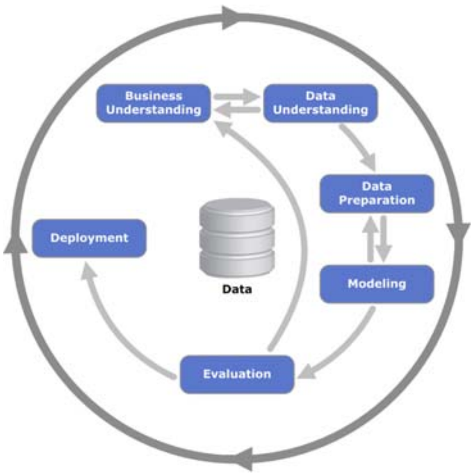

Figure 1: The CRISP-DM process model for data mining.

1 DATA MINING

Data mining – this notion will be used as a synonym for all kinds of smart data analysis – is a complex process that aims at turning raw data into actionable knowledge (see Figure 1 which depicts a standard process model). We will introduce the basic notions, discuss the various steps and in particular have a closer look at the choices to be made and a few pitfalls to be avoided.

In particular, we will address the crucial aspects of how to choose an appropriate modeling approach and how to assess the quality of a solution found by a data analyst.

We show that in many cases it is not a good idea to simply apply the data analyst's favorite modeling technique. Instead we describe the various dimensions of such a choice and encourage the end users of a data analysis to clearly state their requirements.

2 SAMPLE APPLICATIONS

Data Analysis can (and should) play a central role at various stages of the value added chain in the financial industry. In the following we will have a closer look at some relevant activities in this context.

2.1 Appraisal of real economic goods

Scoring and rating processes are at the heart of financial industry. Here we will demonstrate an approach to appraise vessels as typical representatives of real economic goods which form an important class of investments.

2.2 Fraud detection

In B2B scenarios a company's annual accounts form the basis for their credit rating and all further negotiations. Usually, the numbers reported are accepted as a correct representation of last year's business activities. But what if they are manipulated? We describe an approach that identifies abnormalities in annual accounts, thus facilitating the detection of intentional manipulations.

2.3 Identifying interesting customers

There are numerous aspects that can make a customer particularly interesting to a company – his/her interest in certain products, credit-worthiness and default risk, churn rate etc. We describe an integrated approach to identify these individuals that reduces the marketing effort required while simultaneously improving the company's insight into their customer base and the quality of customer contact.

In particular, we will see how the modeling technique applied affects the usefulness of the analytical findings.

2.4 Stock selection

From an abstract point of view, selecting a relevant set of stocks is similar to the previous task as it mainly involves segmentation and classification efforts. However, we will see that data preprocessing in this case is significantly more complex and requires some advanced expert knowledge.

3 Perspective

Big data is more than a buzzword – even if it's not the silver bullet for all problems ahead. We will discuss various techniques and attempts to commercially make use of huge, largely unstructured data sets and briefly discuss potential future applications.

[1] mineway GmbH, Saarbrücken, Germany, email: mbauer@mineway.de

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 $CS_1 : \{r_1, r_3\}$.

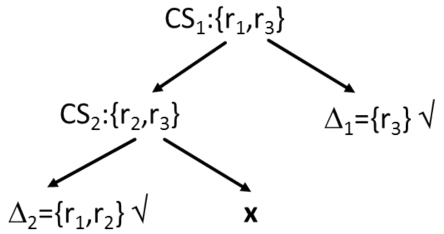

Figure 1. Hitting Set Directed Acyclic Graph (HSDAG) for requirements $CREQ = \{r_1 : av = high, r_2 : wr = low, r_3 : rr = high\}$.

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.

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

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*: $\Delta_1 = \{r_3\}$.

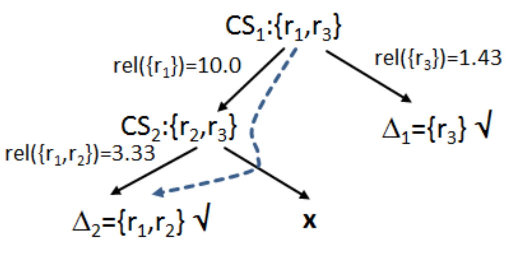

Figure 2. Personalized diagnosis determined for $CREQ$ and the individual importance weights defined in Table 3 (for customer1). In this example, Δ_2 is the preferred diagnosis since $relevance(\Delta_2) > relevance(\Delta_1)$.

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.

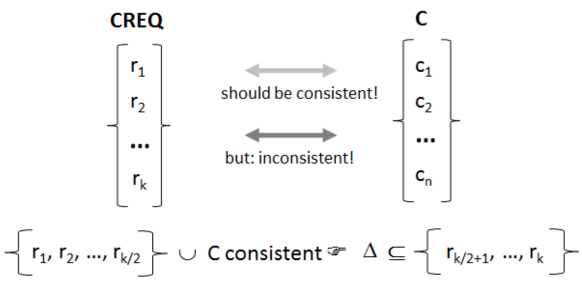

Figure 3. FASTDIAG approach to diagnosis determination. $CREQ$ represents a set of customer requirements and C represents a set of constraints. The algorithm is based on a divide-and-conquer approach: if $\{r_1, r_2, ..., r_{k/2}\}$ is consistent with C then diagnosis search can be continued in $\{r_{k/2+1}...r_k\}$.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

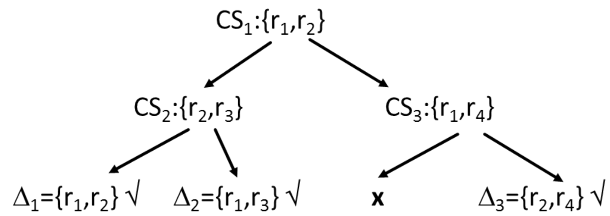

Figure 4. Hitting Set Directed Acyclic Graph (HSDAG) for requirements $CREQ = \{r_1 : rr \geq 5.5, r_2 : rt = 3.0, r_3 : acc = yes, r_4 : bc = yes\}$.

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.

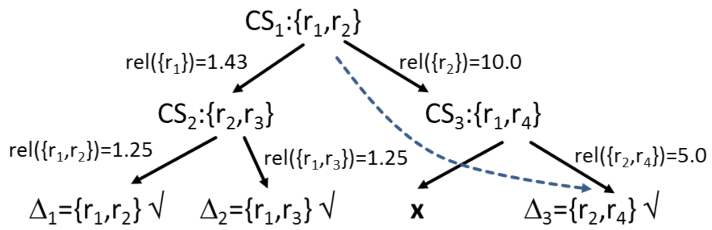

Figure 5. Personalized diagnoses determined for $CREQ$ and the individual importance weights defined in Table 7 (for customer 1). In this example, Δ_3 is the preferred diagnosis.

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 = $\{c_1 : ccw \leq cw, c_2 : ils \leq ls, c_3 : irt = rt, c_4 : pir \geq ir, c_5 : see\ below, c_{6,7} : see\ below\}$

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.

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

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

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).

An Integrated Knowledge Engineering Environment for Constraint-based Recommender Systems

Stefan Reiterer[1]

Abstract. Constraint-based recommenders support customers in identifying relevant items from complex item assortments. In this paper we present a constraint-based environment already deployed in real-world scenarios that supports knowledge acquisition for recommender applications in a MediaWiki-based context. This technology provides the opportunity do directly integrate informal Wiki content with complementary formalized recommendation knowledge which makes information retrieval for users (readers) easier and less time-consuming. The user interface supports recommender development on the basis of intelligent debugging and redundancy detection. The results of a user study show the need of automated debugging and redundancy detection even for small-sized knowledge bases.

1 Introduction

Constraint-based recommenders support the identification of relevant items from large and often complex assortments on the basis of an explicitly defined set of recommendation rules [3]. Example item domains are digital cameras and financial services [5, 8, 9]. For a long period of time the engineering of recommender knowledge bases (for constraint-based recommenders) required that knowledge engineers are technical experts (in the majority of the cases computer scientists) with the needed technical capabilities [14]. Developments in the field moved one step further and provided graphical engineering environments [5], which improve the accessibility and maintainability of recommender knowledge bases. However, users still have to deal with additional tools and technologies which is in many cases a reason for not applying constraint-based environments.

Similar to the idea of Wikipedia to allow user communities to develop and maintain Wiki pages in a cooperative fashion, we introduce the WEEVIS[2] environment, which supports the community-based development of constraint-based recommender applications within a Wiki environment. WEEVIS has been implemented on the basis of MediaWiki[3], which is an established standard Wiki platform. Compared to other types of recommender systems such as collaborative filtering [19] and content-based filtering [25], constraint-based recommender systems are based on an underlying recommendation knowledge base, i.e., recommendation knowledge is defined explicitly. WEEVIS is already applied by four Austrian universities (within the scope of recommender systems courses) and two companies for the purpose of prototyping recommender applications in the financial services domain.

[1] SelectionArts Intelligent Decision Technologies GmbH, Austria, email:stefan.reiterer@selectionarts.com

[2] www.weevis.org.

[3] www.mediawiki.org.

The user interface of the WEEVIS environment provides intelligent mechanisms that help to make development and maintenance operations easier. Based on model-based diagnosis techniques [12, 17, 26], the environment supports users in the following situations: (1) if no solution could be found for a set of user requirements, the system proposes repair actions that help to find a way out from the ”no solution could be found” dilemma; (2) if the constraints in the recommender knowledge base are inconsistent with a set of test cases (situation detected within the scope of regression testing of the knowledge base), those constraints are shown to the users (knowledge engineers) who are responsible for the faulty behavior of the knowledge base; (3) if the recommender knowledge base includes redundant constraints, i.e., constraints that – if removed from the knowledge base – logically follow from the remaining constraints, these constraints are also determined in an automated fashion and shown to knowledge engineers.

The major contributions of this paper are the following. (1) on the basis of a working example from the domain of *financial services*, we provide an overview of the diagnosis and redundancy detection techniques integrated in the WEEVIS environment. (2) we report the results of an empirical study which analyzed the usability of WEEVIS functionalities.

The remainder of this paper is organized as follows. In Section 2 we discuss related work. In Section 3 we present an overview of the recommendation environment WEEVIS and discuss the included knowledge engineering support mechanisms. In Section 4 we present results of an empirical study that show the need of intelligent diagnosis and redundancy detection support. In Section 5 we discuss issues for future work, with Section 6 we conclude the paper.

2 Related Work

Based on original static Constraint Satisfaction Problem (CSP) represenations [15, 20, 29], many different types of constraint-based knowledge representations have been developed. Mittal and Falkenhainer [22] introduced dynamic constraint satisfaction problems where variables have an activity status and only active variables are taken into account by the search process. Stumptner et al. [28] introduced the concept of generative constraint satisfaction where variables can be generated on demand within the scope of solution search. Compared to existing work, WEEVIS supports the solving of static CSPs on the basis of conjunctive queries where each solution corresponds to a result of querying a relational database. Additionally, WEEVIS includes diagnosis functionalities that help to automatically determine repair proposals in situations where no solution could be found [12].

A graphical recommender development environment for single users is introduced in [5]. This Java-based environment supports the development of constraint-based recommender applications for online selling platforms. Compared to Felfernig et al. [5], WEEVIS provides a wiki-based user interface that allows user communities to develop recommender applications. Furthermore, WEEVIS includes efficient diagnosis [12] and redundancy detection [13] mechanisms that allow the support of interactive knowledge base development.

A Semantic Wiki-based approach to knowledge acquisition for collaborative ontology development is introduced in [2]. Compared to Baumeister et al. [2], WEEVIS is based on a recommendation domain specific knowledge representation (in contrast to ontology representation languages) which makes the definition of domain knowledge more accessible also for domain experts. Furthermore, WEEVIS includes intelligent debugging and redundancy detection mechanisms which make development and maintenance operations more efficient. We want to emphasize that intended redundancies can exist, for example, for the purpose of better understandability of the knowledge base. If such constraints are part of a knowledge base, these should be left out from the redundancy detection process.

A first approach to a conflict-directed search for hitting sets in inconsistent CSP definitions was introduced by Bakker et al. [1]. In this work, minimal sets of faulty constraints in inconsistent CSP definitions were identified on the basis of the concepts of model-based diagnosis [26]. In the line of Bakker et al. [1], Felfernig et al. [4] introduced concepts that allow the exploitation of the concepts of model-based diagnosis in the context of knowledge base testing and debugging. Compared to earlier work [4, 24], WEEVIS provides an environment for development, testing, debugging, and application of recommender systems. With regard to diagnosis techniques, WEEVIS is based on more efficient debugging and redundancy detection techniques that make the environment applicable in interactive settings [12, 16, 21].

3 The WEEVIS Environment

In it's current version, WEEVIS supports scenarios where user requirements can be defined in terms of functional requirements [23]. The corresponding recommendations (solutions) are retrieved from a predefined set of alternatives (also denoted as item set or product catalog). Requirements are checked with regard to their consistency with the underlying item set (consistency is given if at least one solution could be identified). If no solution could be found, WEEVIS repair alternatives are determined on the basis of direct diagnosis algorithms [12]. This way, WEEVIS does not only support item selection but also consistency maintenance processes on the basis of intelligent repair mechanisms [6].

WEEVIS is based on the idea that a community of users cooperatively contributes to the development of a recommender knowledge base. The environment supports knowledge acquisition processes on the basis of tags that can be used for defining and testing recommendation knowledge bases. Using WEEVIS, standard Wikipedia pages can be extended with recommendation knowledge that helps to represent domain knowledge in a more accessible and understandable fashion. The same principles used for the developing Wikipedia pages can also be used for the development and maintenance of recommender knowledge bases, i.e., in the *read* mode recommenders can be executed and in the *view source* mode recommendation knowledge can be defined and adapted. This way, rapid prototyping processes can be supported in an intuitive fashion (changes to the knowledge can be immediately experienced by switching from the *view source* to the *read* mode). In the *read* mode, knowledge bases can as well be tested and in the case of inconsistencies (some test cases were not fulfilled within the scope of regression testing) corresponding diagnoses are shown to the user.

3.1 Overview

The website *www.weevis.org* provides a selection of different recommender applications (full list, list of most popular recommenders, and recommenders that have been defined previously) that can be tested and extended. Most of these applications have been developed within the scope of university courses on recommender systems (conducted at four Austrian universities). WEEVIS recommenders can be integrated seamlessly into standard Wiki pages, i.e., informally defined knowledge can be complemented or even substituted with formal definitions.

In the following we will present the concepts integrated in the WEEVIS environment on the basis of a working example from the domain of financial services. In such a recommendation scenario, a user has to specify his/her requirements regarding, for example, the expected capital guarantee level of the financial product or the amount of money he or she wants to invest. A corresponding WEEVIS user interface is depicted in Figure 1 where requirements are specified on the left hand side and the corresponding recommendations are displayed in the right hand side.

Each recommendation (item) has a corresponding support value that indicates the share of requirements that are currently supported by the item. A support value of 100% indicates that each requirement is satisfied by the corresponding item. If the support value is below 100%, corresponding repair alternatives are shown to the user, i.e., alternative answers to questions that guarantee the recommendation of at least one item (with 100% support).

Since WEEVIS is a MediaWiki-based environment, the definition of a recommender knowledge base is supported in a textual fashion on the basis of a syntax similar to MediaWiki. An example of the definition of a (simplified) financial services recommender knowledge base is depicted in Figure 2. Basic syntactical elements provided in WEEVIS will be introduced in the next subsection.

3.2 WEEVIS Syntax

Constraint-based recommendation requires the explicit definition of questions and possible answers, items and their properties, and constraints (see Figure 2).

In WEEVIS the tag &QUESTIONS enumerates the set of user requirements where, for example, *pension* specifies whether the user wants a financial product to support his private pension plan [yes, no] and *maxinvestment* specifies the amout of money the user wants to invest. Furthermore, *payment* represents the frequency in which the payment should be done [once, periodical], *payout* specifies the frequency the customer gets a payout from the financial product (out of [once,monthly]), and *guarantee* the expected capital guarantee [low, high].

An item assortment can be specified in WEEVIS using the &PRODUCTS tag (see Figure 2). In our example, the item (product) assortment is specified by values related to the attributes *name*; *guaranteep*, the capital guarantee the product provides; *payoutp*, the payout frequency of the product; *mininvestp* the minimal amount of

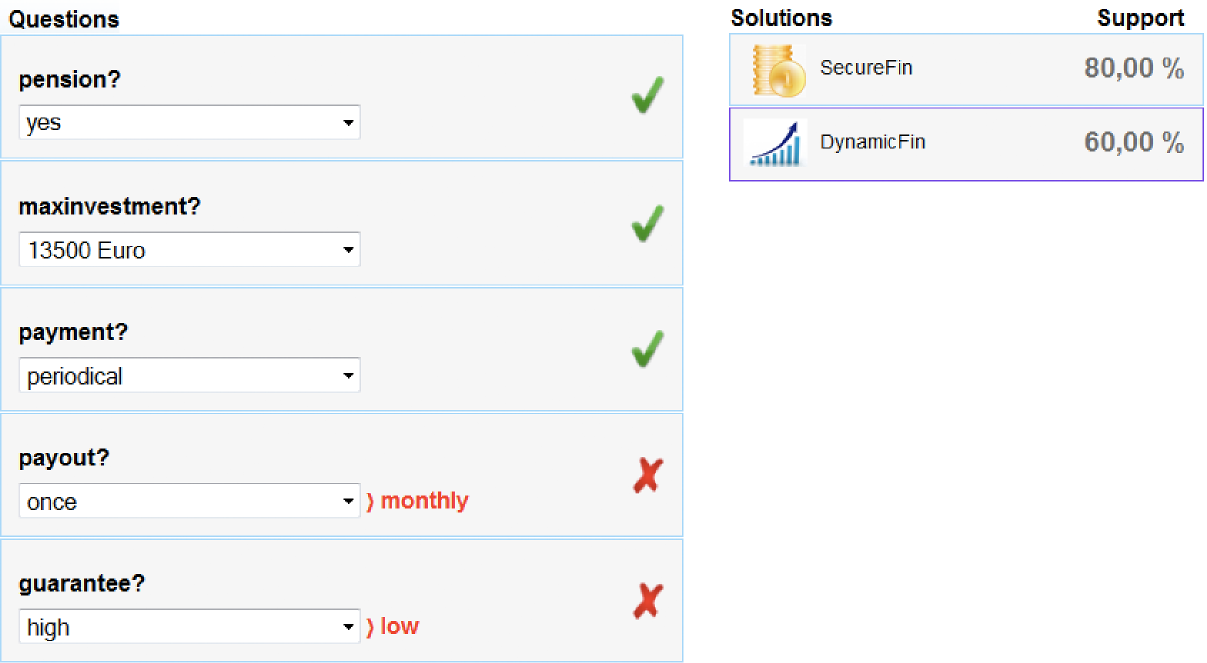

Figure 1. A simple *financial service* recommender (WEEVIS *read* mode).

money for the financial service. Three items are specified: *SecureFin*, *BonusFin*, and *DynamicFin*.

Incompatibility constraints describe incompatible combinations of requirements. Using the *&INCOMPATIBLE* keyword, we are able to describe an incompatibility between the variables *pension* and *guarantee*. For example, financial services with *low* guarantee must not be recommended to users interested in a product that supports their private pension plan. Filter constraints describe relationships between requirements and items, for example, *maxinvest* $\geq$ *mininvestp*, i.e., the amount of money the user is willing to invest must exceed the minimal payment necessary for the financial product.

In addition the recommendation knowledge base itself, WEEVIS supports the specification of test cases that can be used for the purposes of regression testing (see also Section 3.4). After changes to the knowledge base, regression tests can be triggered by setting the —*show*— tag, that specifies whether the recommender system user interface should show the status of the test case (satisfied or not).

3.3 Recommender Knowledge Base

Recommendation knowledge can be represented as a CSP [20] with the variables V ($V = U \cup P$) and the constraints $C = COMP \cup PROD \cup FILT$ where $u_i \in U$ are variables describing possible user requirements (e.g., *pension*) and $p_i \in P$ are describing item properties (e.g., *payoutp*). Furthermore, $COMP$ represents incompatibility constraints of the form $\neg X \vee \neg Y$, $PROD$ the products with their attributes in disjunctive normal form (each product is described as a conjunction of individual product properties), and $FILT$ the given filter constraints of the form $X \rightarrow Y$.

The knowledge base specified in Figure 2 can be translated into a corresponding CSP where &QUESTIONS represents U, &PRODUCTS represents P and $PROD$, and &CONSTRAINTS represents $COMP$ and $FILT$. On the basis of such a definition, WEEVIS is able to calculate recommendations that take into account a specified set of requirements. Such requirements are represented as unary constraints (in our case $R = \{r_1, r_2, ..., r_k\}$).

If requirements $r_i \in R$ are inconsistent with the constraints in C, we are interested in a subset of these requirements that should be adapted in order to be able to restore consistency. On a formal level we define a *requirements diagnosis task* and a corresponding *diagnosis* (see Definition 1).

Definition 1 (Requirements Diagnosis Task). Given a set of requirements R and a set of constraints C (the recommendation knowledge base), the requirements diagnosis task is to identify a minimal set Δ of constraints (the diagnosis) that has to be removed from R such that $R - \Delta \cup C$ is consistent.

An example of a set of requirements inconsistent with the defined recommendation knowledge is $R = \{r_1 : pension = yes, r_2 : maxinvest = 13500, r_3 : payment = periodical, r_4 : payout = once, r_5 : guarantee = high\}$. The recommendation knowledge base induces two minimal conflict sets (CS) [18] in R which are $CS_1 : \{r_1, r_5\}$ and $CS_2 : \{r_1, r_4\}$. For these conflict sets we have two diagnoses: $\Delta_1 : \{r_4, r_5\}$ and $\Delta_2 : \{r_1\}$. The pragmatics, for example, of Δ_1 is that at least r_4 and r_5 have to be adapted in order to be able to find a solution. How to determine such diagnoses on the basis of a HSDAG (hitting set directed acyclic graph) is shown, for example, in [4].

In interactive settings, where diagnoses should be determined in an efficient fashion [12], hitting set based approaches tend to become too inefficient. The reason for this is that conflict sets [18] have to be determined as an input for the diagnosis process. This was the major motivation for developing and integrating FASTDIAG [12] into the WEEVIS environment. Analogous to QUICKXPLAIN [18], this algorithm is based on a divide-and-conquer based approach that en-

```
A very simple "Financial Service Recommender" recommender.
<recommender>
  &PRODUCTS
    {!name!guaranteep!payoutp!#mininvestp!
     |SecureFin|high|once|10000|
     |BonusFin|low|once|3000|
     |DynamicFin|low|monthly|1000|
     }
  &QUESTIONS
    {|pension?(yes, no)|
     |maxinvestment?#(1000,15000,500, Euro)|
     |payment?(once, periodical)|
     |payout? (once, monthly)|
     |guarantee? (high, low)|}
  &CONSTRAINTS
    {
     |pension? = yes &INCOMPATIBLEWITH guarantee? = high|
     |pension? = yes &INCOMPATIBLEWITH payout? = once|
     |maxinvestment? >= #mininvestp|
     |&IF guarantee? = high &THEN guaranteep = high|
     |&IF guarantee? = low &THEN guaranteep = low|
     |&IF guarantee? = high &THEN guaranteep <> low|
     |&IF payout? = once &THEN payoutp = once|
     |&IF payout? = monthly &THEN payoutp = monthly|}
  &TEST
    {|show| pension? = yes, guarantee? = low, payout? = once|}
</recommender>
```

Figure 2. Financial services knowledge base (*view source (edit)* mode).

ables the determination of minimal diagnoses without the determination of conflict sets. A minimal diagnosis Δ can be used as basis for determining repair actions, i.e., concrete measures to change user requirements in R such that the resulting R' is consistent with C.

3.4 Diagnosis and Repair of Requirements

Definition 2 (Repair Task). Given a set of requirements $R = \{r_1, r_2, ..., r_k\}$ inconsistent with the constraints in C and a corresponding diagnosis $\Delta \subseteq R$ ($\Delta = \{r_l, ..., r_o\}$), the corresponding repair task is to determine an adaption $A = \{r'_l, ..., r'_o\}$ such that $R - \Delta \cup A$ is consistent with C.

In WEEVIS, repair actions are determined conform to Definition 2. For each diagnosis Δ determined by FASTDIAG (currently, the first n=3 leading diagnoses are determined), the corresponding solution search for $R - \Delta \cup C$ returns a set of alternative repair actions (represented as adaptation A). In the following, all products that satisfy $R - \Delta \cup A$ are shown to the user (see the right hand side of Figure 1).

Diagnosis determination in FASTDIAG is based on a total lexicographical ordering of the customer requirements [12]. This ordering is derived from the order in which a user has entered his/her requirements. For example, if $r_1 : pension = yes$ has been entered before $r_4 : payout = once$ and $r_5 : guarantee = high$ then the underlying assumption is that r_4 and r_5 are of lower importance for the user and thus have a higher probability of being part of a diagnosis. In our working example $\Delta_1 = \{r_4, r_5\}$. The corresponding repair actions (solutions for $R - \Delta_1 \cup C$) is $A = \{r'_4 : payout = monthly, r'_5 : guarantee = low\}$, i.e., $\{r_1, r_2, r_3, r_4, r_5\} - \{r_4, r_5\} \cup \{r'_4, r'_5\}$ is consistent. The item that satisfies $R - \Delta_1 \cup A$ is $\{DynamicFin\}$ (see in Figure 2). The identified items (p) are ranked according to their support value (see Formula 1).

$$support(p) = \frac{\#adaptions\ in\ A}{\#requirements\ in\ R} \tag{1}$$

3.5 Regression Testing

WEEVIS supports regression testing processes by the definition and execution of (positive) test cases which specify the intended behavior of the knowledge base. If some of the test cases are not accepted by the knowledge base (are inconsistent with the knowledge base), the causes of this unintended behavior have to be identified. On a formal level a *recommender knowledge base (RKB) diagnosis task* can be defined as follows (see Definition 3).

Definition 3 (RKB Diagnosis Task). Given a set C (recommender knowledge base) and a set $T = \{t_1, t_2, ..., t_q\}$ of test cases t_i , the diagnosis task is to identify a minimal set Δ of constraints (the diagnosis) that have to be removed from C such that $\forall t_i \in T : C - \Delta \cup \{t_i\}$ is consistent.

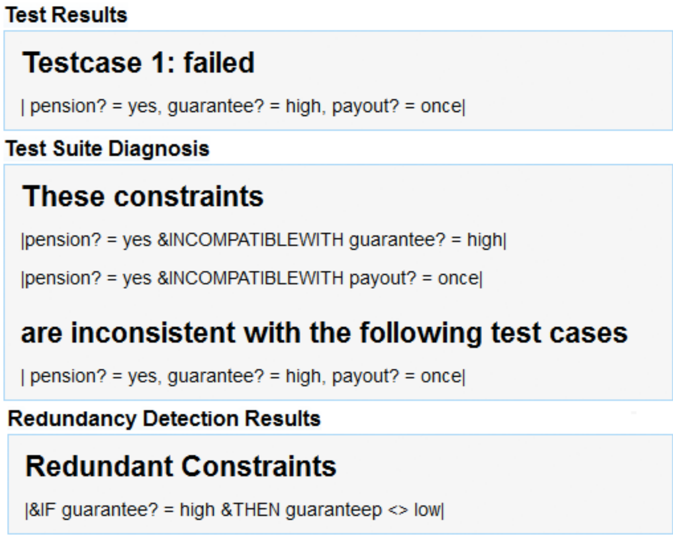

Figure 3. WEEVIS maintenance support: diagnosis and redundancy detection.

An example test case inducing an inconsistency with C is t : $pension = yes$ and $guarantee = high$ and $payout = once$ (see Figure 2). In this context, t induces two conflicts in C which are $CS_1 : \neg(pension = yes \wedge guarantee = high)$ and $CS_2 : \neg(pension = yes \wedge payout = once)$. In order to make C consistent with t, both incompatibility constraints have to be deleted from C, i.e., are part of the diagnosis Δ (see Figure 3).

In contrast to the hitting set based approach [4], WEEVIS includes a FASTDIAG based approach for knowledge base debugging which is more efficient and can therefore be applied in interactive settings [12]. In this context, diagnoses are searched in C (the test cases used for regression testing are assumed to be correct). In the case of requirements diagnosis, the total ordering of the requirements is related to user preferences. In the case of knowledge base diagnosis [4, 16], the ordering is currently derived from the ordering of the constraints in the knowledge base.

3.6 Identifying Redundancies

To support users in identifying redundant constraints in recommender knowledge bases, the COREDIAG [13] algorithm has been integrated into the WEEVIS environment. COREDIAG relies on QUICKXPLAIN [18] and is used for the determination of minimal cores (minimal non-redundant constraint sets). On a formal level a *recommendation knowledge base (RKB) redundancy detection task* can be defined as follows (see Definition 4).

Definition 4 (RKB Redundancy Detection Task). Let c_a be a constraint of C (the recommendation knowledge base) and $\overline{C}$ the logical negation (the complement or inversion) of C. Redundancy can be analyzed by checking $C - \{c_a\} \cup \overline{C}$ for consistency - if consistency is given, c_a is non-redundant. If this condition is not fulfilled, c_a is said to be redundant. By iterating over each constraint of C, executing the non-redundancy check $C - \{c_a\} \cup \overline{C}$, and deleting redundant constraints from C results in a set of non-redundant constraints (the minimal core).

As an example, the knowledge base shown in Figure 2 contains redundancies. Consequently, the corresponding set of constraints C does not represent a minimal core. Taking a closer look at the knowledge base it appears that two individual filter constraints are redundant with each other. More precisely, either the constraint *&IF guarantee? = high &THEN guaranteep = high* or the constraint *&IF guarantee? = high &THEN guaranteep <> low* can be removed from the knowledge base (in our example, the latter is proposed as redundant by COREDIAG – see Figure 3). In the general case, higher cardinality constraint sets can be removed, not only cardinality-1 sets as in our example [13].

Similar to the diagnosis of inconsistent requirements the COREDIAG algorithm is based on the principle of divide-and-conquer: whenever a set S which is a subset of C is inconsistent with $\overline{C}$, it is or contains a minimal core, i.e., a set of constraints which preserve the semantics of C. COREDIAG is based on the principle of QUICKXPLAIN [18]. As a consequence a minimal core (minimal set of constraints that preserve the semantics of C) can be interpreted as a minimal conflict, i.e., a minimal set of constraints that are inconsistent with $\overline{C}$. Based on the assumption of a strict lexicographical ordering [12] of the constraints in C, COREDIAG determines preferred minimal cores.

4 Empirical Study

4.1 Study Design

We conducted an experiment to highlight potential reductions of development and maintenance efforts facilitated by the WEEVIS debugging and redundancy detection support. For this study we defined four knowledge bases that differed with regard to the number of constraints, variables, faulty constraints, and redundancies (see Table 1). Based on these example knowledge bases, the participants had to find solutions for the following two types of tasks:

1. *Diagnosis task*: The participants had to answer the question which minimal set Δ of faulty constraints has to be removed from C ($C = COMP \cup FILT$) such that there exists at least one solution for ($(C - \Delta) \cup PROD$).

2. *Redundancy detection task*: The participants had to answer the question which constraints in $C = COMP \cup FILT$ are redundant (if $C - \{c_a\} \cup \overline{C}$ is inconsistent then the constraint c_a is redundant).

knowledge base	number of constraints /variables /faulty constraints /test cases /redundancies
$kb_1(redundant)$	5/5/0/0/2
$kb_2(inconsistent)$	5/5/1/2/0
$kb_3(redundant)$	10/10/0/0/4
$kb_4(inconsistent)$	10/10/2/4/0

Table 1. Knowledge bases used in the empirical study.

The participants (subjects N=20) of our experiment were separated into two groups (groups *A* and *B*). All subjects were students of Computer Science (20% female, 80% male) who successfully completed a course on constraint technologies and recommender systems. Each subject had to complete the assigned tasks on his/her own on a sheet of paper and they had to track the time for each task. In our experiment we randomly assigned the participants to one of the two test groups shown in Table 2. This way we were able to compare the time efforts of identifying faulty constraints and redundancies in knowledge bases as well as to estimate error rates related to the given tasks.

testgroup	1^{st} knowledge base	2^{nd} knowledge base
A ($n = 10$)	kb_1 (redundancy detection)	kb_4 (diagnosis)
B ($n = 10$)	kb_2 (diagnosis)	kb_3 (redundancy detection)

Table 2. Each subject had to complete one diagnosis and one redundancy detection task. Members of group A had a redundancy detection task of lower complexity and a higher complexity diagnosis detection task (randomized order). Vice-versa members of group B had to solve a higher complexity redundancy detection and a lower complexity diagnosis task.

4.2 Study Results

The *first goal* of our experiment was to analyze time efforts and error rates related to the identification of faulty constraints in recommender knowledge bases. The first hypothesis tested in our experiment was the following:

Hypothesis 1: Even low-complexity knowledge bases trigger the identification of faulty diagnoses (note that all knowledge bases used in the experiment can be interpreted as low-complexity knowledge bases [13]).

The average time effort for identifying minimal diagnoses in knowledge base kb_2 was 281.3 seconds, the average time needed to identify diagnoses in kb_4 was 497.5 seconds. The results show a significantly higher error rate when the participants had to identify the faulty constraints in the more complex knowledge base (see Table 3). Hypothesis 1 can be confirmed by the results in Table 3 that show that even simple knowledge bases trigger high error rates and increasing time efforts. With the automated diagnosis detection mechanisms integrated in WEEVIS, reductions of related error rates and time efforts can be expected.

	$groupB$ (kb_2)	$groupA$ (kb_4)
average time (sec.)	281.3	497.5
correct (%)	50.0	10.0
incorrect (%)	50.0	90.0

Table 3. Time efforts and error rates related to the completion of diagnosis tasks.

The *second goal* of our experiment was to analyze time efforts and error rates related to the identification of redundant constraints in recommender knowledge bases. The second hypothesis tested in our experiment was the following:

Hypothesis 2: Even low-complexity knowledge bases trigger the faulty identification of redundant constraints.

The average time for identifying redundant constraints in knowledge base kb_1 was 189.2 seconds, for kb_3 337.4 seconds were needed. The results show a significantly higher error rate when the participants had to identify redundant constraints in the more complex knowledge base (see Table 4). Hypothesis 2 can be confirmed since even for low complexity knowledge bases error rates related to redundancy detection tasks are high. With the automated redundancy detection mechanisms integrated in WEEVIS, reductions of related error rates and time efforts can be expected.

	$groupA$ (kb_1)	$groupB$ (kb_3)
average time (sec.)	189.2	337.4
correct (%)	40.0	0.0
incorrect (%)	60.0	100.0

Table 4. Time efforts and error rates related to the completion of redundancy detection tasks.

5 Future Work

There are a couple of issues for future work. The current WEEVIS version does not include functionalities that allow the learning/prediction of user preferences. The importance of individual user requirements is based on the assumption that the earlier a requirement has been specified the more important it is. In future versions we want to make the modeling of preferences more intelligent by integrating, for example, learning mechanisms that derive requirements importance distributions on the basis of analyzing already completed recommendation sessions.

Diagnoses and redundancies are currently implemented on the level of constraints, i.e., intra-constraint diagnoses and redundancies are not supported. In future WEEVIS versions we want to integrate fine-granular analysis methods that will help to make analysis and repair of constraints even more efficient. A major research challenge in this context is to integrate intelligent mechanisms for diagnosis discrimination [27] since in many scenarios quite a huge number of alternative diagnoses exists. In such scenarios it is important for knowledge engineers to receive recommendations of diagnoses that are reasonable. This challenge has already been tackled in the context of diagnosing inconsistent user requirements (see, e.g., [6]), however, heuristics with high prediction quality for knowledge bases have not been developed up to now [10, 11].

A major issue for future work is to integrate alternative mechanisms for knowledge base development and maintenance. The knowledge engineer centered approach to knowledge base construction leads to scalability problems in the long run, i.e., knowledge engineers are not able to keep up with the speed of knowledge base related change and extension requests. An alternative approach to knowledge base development and maintenance is the inclusion of concepts of Human Computation [7, 30] which allow a more deep integration of domain experts into knowledge engineering processes on the basis of simple micro tasks. Resulting micro contributions can be automatically integrated into constraints part of the recommendation knowledge base.

Finally, we are interested in a better understanding of the key factors that make knowledge bases understandable. More insights and answers related to this question will help us to better identify problematic areas in a knowledge base which could cause maintenance efforts above average. A first step in this context will be to analyze existing practices in knowledge base development and maintenance with the goal to figure out major reasons for the knowledge acquisition bottleneck and how this can be avoided in the future.

6 Conclusion

In this paper we presented WEEVIS which is an open constraint-based recommendation environment. By exploiting the advantages of Mediawiki, WEEVIS provides an intuitive basis for the development and maintenance of constraint-based recommender applications. WEEVIS is already applied by four Austrian universities within the scope of recommender systems courses and also applied by companies for the purpose of prototyping recommender applications. The results of our empirical study indicate the potential of reductions of error rates and time efforts related to diagnosis and redundancy detection. In industrial scenarios, WEEVIS can improve the quality of knowledge representations, for example, documentations can at least partially be formalized which makes knowledge more accessible – instead of reading a complete documentation, the required knowledge chucks can be identified easier.

REFERENCES

[1] R. Bakker, F. Dikker, F. Tempelman, and P. Wogmim, 'Diagnosing and Solving Over-determined Constraint Satisfaction Problems', in *13th International Joint Conference on Artificial Intelligence*, pp. 276–281, Chambery, France, (1993).

[2] J. Baumeister, J. Reutelshoefer, and F. Puppe, 'KnowWE: a Semantic Wiki for Knowledge Engineering', *Applied Intelligence*, **35**(3), 323–344, (2011).

[3] A. Felfernig and R. Burke, 'Constraint-based Recommender Systems: Technologies and Research Issues', in *10th International Conference on Electronic Commerce*, p. 3. ACM, (2008).

[4] A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner, 'Consistency-based Diagnosis of Configuration Knowledge Bases', *Artificial Intelligence*, **152**(2), 213–234, (2004).

[5] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, 'An Integrated Environment for the Development of Knowledge-based Recommender Applications', *International Journal of Electronic Commerce*, **11**(2), 11–34, (2006).

[6] A. Felfernig, G. Friedrich, M. Schubert, M. Mandl, M. Mairitsch, and E. Teppan, 'Plausible Repairs for Inconsistent Requirements', in *IJCAI*, volume 9, pp. 791–796, (2009).

[7] A. Felfernig, S. Haas, G. Ninaus, M. Schwarz, T. Ulz, M. Stettinger, K. Isak, M. Jeran, and S. Reiterer, 'RecTurk: Constraint-based Recommendation based on Human Computation', in *RecSys 2014 CrowdRec Workshop*, pp. 1–6, Foster City, CA, USA, (2014).

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

[9] 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).

[10] A. Felfernig, S. Reiterer, M. Stettinger, and J. Tiihonen, 'Intelligent Techniques for Configuration Knowledge Evolution', in *VAMOS Workshop 2015*, pp. 51–60, Hildesheim, Germany, (2015).

[11] A. Felfernig, S. Reiterer, M. Stettinger, and J. Tiihonen, 'Towards Understanding Cognitive Aspects of Configuration Knowledge Formalization', in *VAMOS Workshop 2015*, pp. 117–124, Hildesheim, Germany, (2015).

[12] A. Felfernig, M. Schubert, and C. Zehentner, 'An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **26**, 53–62, (2012).

[13] A. Felfernig, C. Zehentner, and P. Blazek, 'COREDIAG: Eliminating Redundancy in Constraint Sets', *International Workshop on Principles of Diagnosis (DX'11)*, 219–224, (2011).

[14] G. Fleischanderl, G. Friedrich, A. Haselböck, H. Schreiner, and M. Stumptner, 'Configuring Large Systems Using Generative Constraint Satisfaction', *IEEE Intelligent Systems*, **13**(4), 59–68, (1998).

[15] E. Freuder, 'In Pursuit of the Holy Grail', *Constraints*, **2**(1), 57–61, (1997).

[16] G. Friedrich, 'Interactive Debugging of Knowledge Bases', in *International Workshop on Principles of Diagnosis (DX'14)*, pp. 1–4, Graz, Austria, (2014).

[17] Russell Greiner, Barbara A. Smith, and Ralph W. Wilkerson, 'A Correction to the Algorithm in Reiter's Theory of Diagnosis', *Artificial Intelligence*, **41**(1), 79–88, (1989).

[18] U. Junker, 'QUICKXPLAIN: Preferred Explanations and Relaxations for Over-constrained Problems', in *AAAI*, volume 4, pp. 167–172, (2004).

[19] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, 'GroupLens: Applying Collaborative Filtering to Usenet News', *Communications of the ACM*, **40**(3), 77–87, (1997).

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

[21] 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).

[22] S. Mittal and B. Falkenhainer, 'Dynamic Constraint Satisfaction', in *National Conference on Artificial Intelligence*, pp. 25–32, (1990).

[23] S. Mittal and F. Frayman, 'Towards a Generic Model of Configuraton Tasks', in *IJCAI*, volume 89, pp. 1395–1401, (1989).

[24] B. O'Sullivan, A. Papadopoulos, B. Faltings, and P. Pu, 'Representative Explanations for Over-constrained Problems', in *Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07)*, eds., R. Holte and A. Howe, pp. 323–328, Vancouver, Canada, (2007). AAAI Press.

[25] M. Pazzani and D. Billsus, 'Learning and Revising User Profiles: The Identification of Interesting Web Sites', *Machine learning*, **27**(3), 313–331, (1997).

[26] R. Reiter, 'A Theory of Diagnosis From First Principles', *Artificial intelligence*, **32**(1), 57–95, (1987).

[27] K. Shchekotykhin and G. Friedrich, 'Diagnosis discrimination for ontology debugging', in *ECAI 2010*, pp. 991–992, (2010).

[28] M. Stumptner, G. Friedrich, and A. Haselböck, 'Generative Constraint-based Configuration of Large Technical Systems', *AI EDAM*, **12**(04), 307–320, (1998).

[29] E. Tsang, *Foundations of Constraint Satisfaction*, volume 289, Academic press London, 1993.

[30] L. VonAhn, 'Human Computation', in *Technical Report CM-CS-05-193*, (2005).

A Personal Data Framework for Exchanging Knowledge about Users in New Financial Services

Beatriz San Miguel, Jose M. del Alamo and **Juan C. Yelmo**[1]

Abstract. Personal data is a key asset for many companies, since this is the essence in providing personalized services. Not all companies, and specifically new entrants to the markets, have the opportunity to access the data they need to run their business. In this paper, we describe a comprehensive personal data framework that allows service providers to share and exchange personal data and knowledge about users, while facilitating users to decide who can access which data and why. We analyze the challenges related to personal data collection, integration, retrieval, and identity and privacy management, and present the framework architecture that addresses them. We also include the validation of the framework in a banking scenario, where social and financial data is collected and properly combined to generate new socio-economic knowledge about users that is then used by a personal lending service.

1 INTRODUCTION

Tailored and customized features are increasingly becoming more popular in IT services. These adjust offers and functionalities of services to the user preferences, interests and personal needs, generally going beyond functionality of the service itself and thus, improving it. In the banking sector, it is not an exception and for some time now new players have appeared to offer financial services based on personalization and recommendations.

Traditionally, banks have been early adopters of new technology solutions, but mainly following a bank-centric approach that users are rarely able to notice [1]. IT companies and new service providers have leveraged this gap to offer user-centric financial services. For example, on-line payment is one of the most competitive areas into which IT companies such as PayPal, Google or Apple, have entered. Moreover, many financial services related to crowdfunding, lending clubs, investment recommendations, financial aggregators that allow the management of personal finances, the comparison or recommendation of banking products, etc. have transformed the traditional ways of financial organizations, or have even created entirely new ones.

These innovative financial services create new opportunities, but also potential threats in the industry. It is vital for banks to understand the new directions and develop threats into new opportunities and returns. In this sense, most of these new financial services require personal data and financial information about users in order to know them better and then, offer and improve services. Here banks possess inherent competitive advantages, since they have a large amount of customer data, transaction information, and the capabilities to enable financing and secure services [2] and [3].

Well aware of this situation, in 2014 the Center for Open Middleware (COM), a joint technology center created by Santander Bank and Universidad Politécnica de Madrid, launched a pilot project intended to research, analyze and evaluate new potential opportunities and applications around personal data. Specifically, the project aims to establish a framework that allows the sharing and use of personal data among companies, and the creation of knowledge about users, while allowing users to manage and control their flow of personal information, defining who access which data and why.

In this paper we introduce the aforementioned framework which has been called the Personal Data Framework (PeDF). The PeDF includes mechanisms for gaining access to personal data from several heterogeneous data sources, and integrating them to facilitate their analysis and processing to produce and infer new knowledge about users. This information can be provided to new financial service providers that, as new players, do not have sufficient personal data to offer their services. On the other hand, there are currently tensions related to the use of personal data, causing privacy and trust concerns in users. In this context, the European public sector is attempting to regulate and evolve the existing legislation to strengthen individual rights in relation to the uses of their personal data and their privacy, while boosting digital and personal data economy [4]. Therefore, the framework includes the necessary tools to involve users in the management and control of their personal information.

The remainder of the paper is organized as follows. First, Section 2 includes the technological background for each issue that covers the PeDF related to personal data: collection, integration, retrieval, and identity and privacy management. Then, Section 3 describes the PeDF architecture, and Section 4 includes the PeDF validation that we have conducted in the financial context. Finally, we present related work in Section 5, and conclude the paper by highlighting conclusions and future directions in Section 6.

2 TECHNOLOGICAL APPROACCHES

The PeDF acts as an intermediate entity between service providers and individuals to allow the former to share and exchange existing personal data and new knowledge obtained from them which cannot be done unilaterally, while enabling users to retrieve a global view of their personal information and decide who can access which data and why. To make it possible, the PeDF has to include mechanisms for gaining access to personal data that are scattered across different service providers (data sources). When the data sources supply personal data to the PeDF, it has to be able to integrate them. This integration must allow the PeDF to provide

[1] Center for Open Middleware, Universidad Politécnica de Madrid, Spain, email: {beatriz.sanmiguel, jose.delalamo, juancarlos.yelmo}@centeropenmiddleware.com

personal data and knowledge obtained from these data to service providers (referred to as data consumers). All of the above has to be controlled by the user and thus, it requires the PeDF to include identity and privacy management solutions.

In summary, the PeDF covers four main technological issues: personal data collection, integration, retrieval, and identity management and privacy. Next, we will present the background associated with each issue, detailing its technological solutions.

2.1 Personal data collection

Data sources can be classified into two main categories in relation to personal data access: public or private, but one source can be categorized as both, depending on the personal data concerned.

The public data sources contain personal data that are accessible in an equitable way for any entity in the public network. On the other hand, in the private data sources, the personal data can only be accessed by authorized entities. We can think of numerous examples of personal data sources, such as social networks, instant messaging services, mobile applications, and many other service providers specialized in a specific user domain such as education, banking, or e-commerce. As an illustrative example, a social network can act as a public or private data source depending on the user configuration.

There are different technologies that allow third parties to collect the personal data from data sources. For the public ones, the so-called Internet bots, spiders, or web crawlers are the most representative. These are software solutions that automatically search, access and retrieve public information on the Internet.

As regards private data sources, there are several mechanisms based on user consent that allow third parties to access the protected personal data. One of the easiest ways is the method based on data files. This kind of files contains personal data created by a user in a specific data source and can be exported by users. For example, Google allows its users to access their personal data, downloading different files[2]. The main problem associated with this solution is that it requires extra work for the users, since they have to be actively involved to download their files, carrying out manual tasks. Moreover, files can be easily manipulated to change their content, and therefore, the security mechanisms are weak. In order to solve this problem, a set of programming functions, protocols, and standards has appeared to automate the process: data sharing Application Programming Interfaces (APIs).

APIs have become the de facto mechanism for sharing and exchanging personal data, since they allow different software applications to communicate and interact directly [3]. They offer code-based access to different functionalities and services to third parties by abstracting their implementation details. On the Internet, the Representational State Transfer (REST) [5] architectural style has recently emerged as the favorite for implementing APIs. It is based on the Hypertext Transfer Protocol (HTTP) to allow connectivity, but it does not specify the syntax of messages. The individual messages and interfaces are designed according to the suppliers' semantic. For example, Facebook and Twitter include different APIs (Graph API[3] and REST APIs[4], respectively) to read and write their user personal data, which are based on the HTTP for communication, and JavaScript Object Notation (JSON) [6] for data interchange. Although the same protocol and language still apply, there are differences, since the suppliers' API use different syntax and semantic to refer to the same data.

In a nutshell, there is no unified API specification, each API contains its own description, which can be poorly documented, and therefore, understanding each one is challenging. There are some initiatives to solve the associated API problems, such as the OpenSocial standards [7] that include a set of open APIs that developers can use to gain access to user personal resources hosted by different providers who have implemented them. We can find a few related solutions in the social network services, such as [8], that proposes a framework to integrate the interaction with different social APIs.

2.2 Personal data integration

Data integration is an old field of research that aims at combining data from different sources and providing them in a unified view [9]. Over time, many solutions have been proposed [10], but two main approaches regarding storage can be followed:

- Centralized way. The personal data is retrieved from external data sources, saved, and stored in a central repository. This is a replication of the personal data stored by data sources and thus, maintaining and updating the replicated data is a key issue. It must incorporate techniques to carry out a periodical refreshing of personal data, or even better, mechanisms that allow the detection of data changes in real time. Despite the aforementioned, it has clear benefits related to availability and timeliness. Furthermore, it facilitates data analysis and processing.
- Decentralized way. Here, there is a central directory or registry and a distributed data storage. It entails little or no storage since personal data is maintained and stored by each external data source. However, personal data access is more complex and generally less efficient than the previous way because recovering data is carried out on the fly and there can be source access limitations.

The two mechanisms are complementary since the central repository of the first way can be considered as an extra storage point for the decentralized solution. Furthermore, both solutions face the challenges of corresponding personal data at different data sources, and giving them a common definition. The former entails the development of algorithms and mapping techniques that (semi)automate the correspondence process to eliminate manual tasks. On the other hand, the common definition of personal data involves establishing a standard to represent the personal data.

There is no standard or a generally adopted representation for personal data, neither the structure (format of the representation), nor even the semantic (meaning of the content). We can find many proposals for standards and proprietary solutions to define each personal data category, almost as many as there are service providers. One of the most promising solutions for integrating all these discrepancies is the use of ontologies.

An ontology is an engineering artifact made up of a vocabulary that describes a certain reality, and a set of explicit assumptions regarding the intended meaning of the vocabulary terms [11]. It enables a common understanding of a specific domain to be shared across a wide range of service providers, adding interoperability, consistency, reusability, and many other advantages [12].

[2] https://support.google.com/accounts/answer/3024190?hl=en
[3] https://developers.facebook.com/docs/graph-api
[4] https://dev.twitter.com/rest/public

Over time, many ontologies have been proposed for diverse domains including healthcare, molecular biology, or web searching. There are general ontologies describing concepts (e.g., object, process and event) that are the same across different domains, such as the Suggested Upper Merged Ontology (SUMO) [13]. Additionally, there are more specific ontologies (namely domain ontologies) that represent the particular concepts of a domain. In the social network field, the Friend of a Friend (FOAF) ontology [14] includes the main terms to describe people, the links between them and the things they create and do on Internet. In the financial industry, the Financial Industry Business Ontology (FIBO) [15] is an ongoing definition of financial industry terms such as contracts, product/service specifications and governance compliance documents. SUMO also includes domain ontologies for finance and economy.

Finally, there are different methodologies and languages for defining your own ontologies, such as those described in [16]. One of the most popular languages is the Web Ontology Language (OWL) [18] that is part of the W3C technology stack. OWL allows the definition of concepts and the complex and rich relationships between them.

2.3 Personal data and knowledge retrieval

Personal data can be offered to third entities, and even more interestingly, these data can be analyzed and processed to obtain knowledge that cannot be achieved unilaterally by service providers. The process for producing this knowledge is referred to as user modelling in the literature [19].

Traditionally, user modelling is a one-sided process in which service providers autonomously collect personal data and then generate user models that satisfy their business needs in a specific domain. A user model is understood as the interpretation of a person in a specific context for an organization. It includes what the organization thinks the user is, prefers, wants, or is going to do, and comprises mainly derived and inferred data. The user model can be used to recommend new contents or services, personalize user interaction, or predict user behavior, among others.

There are different techniques to create user models, choosing one or another depends on what information is been stored and the final application of the model. Next, we point out some of the approaches that can be taken.

2.3.1 Vector-based models

Here, a user is represented by a set of feature-value pairs. The features can be items or concepts of a domain, such as products of a shop, or links on a web site. Each of them has associated a value (usually, a boolean or real number) that indicates the attitude of a user to this feature. For example, the value can indicate whether a user has searched for a product or the number of visits to a link.

There are other approaches similar to this one such as keyword-based, bag of words, or user-items rating matrix [20], which consider only words or terms interesting to users with or without an associated value, or historical user ratings on items, respectively.

This approach is one of the simplest since its implementation and retrieval is quite easy. It has been used by nearly every information retrieval system [21]. However, it is difficult to share with other data consumers because the features and values can be misinterpreted. Moreover, there is a lack of connection between concepts and it does not help in modelling users for other contexts.

2.3.2 Stereotypes

Stereotype modelling [21] attempts to cluster all possible users of a system into different groups, namely stereotypes. Each user that belongs to the same stereotype is treated like the rest of the members of the group so his or her individual features are not considered. Typically, the data used in the classification is a demographic that users have to provide, for example in a registration form.

The main goals of this modelling approach are to define the stereotypes of a system and to implement the trigger techniques that provide mapping from a specific user to one stereotype. These include different clustering analyses, machine-learning techniques and reasoning among others [22]. There is an obvious disadvantage of this approach and it lies in the limited personalization and individualization of users, besides the difficulty in recovering new user models from the existing ones.

2.3.3 Classifier based models

Classifier systems [23] use information about items or the domain together with user data as an input to generate a custom response to the user. These can be implemented using different machine learning methods and the user model is represented as the particular model structure of the used classifier. For example, there can be user models based on decision trees, association rules, or Bayesian Networks. This approach, like the previous ones, has difficulties in retrieving and sharing user models since it is very limited and is based on solving specific tasks.

2.3.4 Semantic user modelling

Semantic technologies have appeared as a way to solve communication problems, and interoperability issues among systems, and to provide and facilitate reusability, reliability, and a common specification [12]. Semantic user modelling [20] is based on using ontologies that model a user or a specific domain using a rich network where terms are connected by different kinds of links that indicate its relations [24].

Using ontologies solves the polysemy problem and facilitates to retrieve and share user models between entities. There are different languages and techniques that allow the extraction of data from ontologies. For example, the SPARQL Protocol and Resource Description Framework (RDF) Query Language (SPARQL) and the accompanying protocols [25] make possible to send queries and receive results from semantic data (expressed as RDF information), e.g., through HTTP. Moreover, new relations between concepts and thus, about user features, can be inferred from ontology representation. Particularly, reasoner engines [16] are software components that allow autonomously the discovery of new knowledge from ontologies. Generally, they employ their own rules, axioms and appropriate chaining methods. We can find stand-alone reasoners, such as Pellet[5], or reasoners included in different semantic frameworks as for example, Protégé[6] and Jena[7].

[5] http://clarkparsia.com/pellet
[6] http://protege.stanford.edu/
[7] https://jena.apache.org/

2.4 Identity Management and Privacy

Identity management commonly refers to the processes involved in the management and selective disclosure of personal data, either within an institution or between several entities, while preserving and enforcing both privacy and security requirements. There are different approaches to implementing identity management, mainly: network-centric and user-centric approaches [26].

Network-centric approaches are based on agreements between service providers that establish trust relationships. Each service provider maintains its own personal data but users can link (federate) isolated accounts that they own across different providers to be recognized within the federated domain. Technological standards for identity federation include the OASIS Security Assertion Markup Language (SAML) [27] and the Kantara Initiative[8].

On the other hand, user-centric approaches highlight user empowerment in the governing of their personal information. Generally, there is a third entity that is in charge of providing user identity to service providers and the user is in the center of the transactions, managing the sharing of personal data. Examples of this approach are [28]: OpenID, OAuth 2.0, and OpenID Connect. Most of the social-based APIs for personal information sharing rely on OAuth 2.0, as for example the Facebook Login API[9]. It introduces a third role to the traditional client-server authentication/authorization model: the resource owner. Following this model, the client (who is not the resource owner, but is acting on his behalf) requests access to resources controlled by the resource owner, but hosted by a container i.e. the online social network. OAuth 2.0 allows the service provider to verify the identity of the client making the request, as well as ensuring that the resource owner has authorized the transaction without revealing their credentials.

Identity management technologies also contribute to privacy management by allowing users to decide on the sharing process. However, this is not enough, as any system managing personal information must abide by the privacy and data protection legal framework in place, and thus fulfill a set of requirements derived from the legal principles. For example, in Europe the main principles include lawfulness collection and processing; gathering specific, informed and explicit consent from data subjects; purpose binding; necessity and data minimization; transparency and openness; rights of the individual; and, security safeguards [29].

The state of the art includes a plethora of technological solutions, each addressing a specific privacy concern, and globally referred to as Privacy Enhancing Technologies (PETs) [29]. However, adding PETs on top of an existing system does not solve all privacy requirements, and thus there is a general consensus on the need to introduce Privacy by Design (PbD) approaches when developing systems i.e. considering privacy issues from the onset of a project and through its entire lifecycle [30].

All the aforementioned technologies facilitate the access and management of personal data. However, user-centric solutions allow users to control and manage their personal data directly, bringing a better user-experience.

[8] https://kantarainitiative.org/
[9] https://developers.facebook.com/products/login/

3 FRAMEWORK ARCHITECTURE

As described in the previous section, there are many solutions and specific technologies to handle the design and implementation of the PeDF. We have proposed a comprehensive architecture for the PeDF that considers different approaches for personal data collection, integration, retrieval, and identity and privacy management, regardless of the specific technologies and implementations. Figure 1 represents this PeDF architecture where we can distinguish its modules, and its relationships with different external data sources, data consumers, and the user.

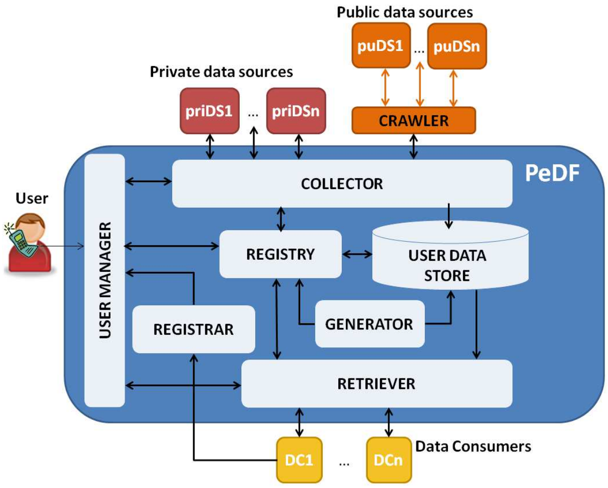

Figure 1. Personal Data Framework architecture

Firstly, we have considered that there are diverse existing data sources (private or public), and crawlers on the Internet that can be linked with the PeDF to gain access to user personal data. This data source-user association can be carried out by the user through the User Manager module, or by data consumers via the Registrar module but the latter requires user consent.

Once the data sources are linked, the Collector module is in charge of obtaining personal data from them and these data have to be integrated. We have proposed two complementary approaches to carry out this integration. One is based on collecting and storing personal data, which requires a User Data Store module. The other method is based on indexing personal data, which entails a Registry module that identifies which personal data can be accessed and where they are stored.

Moreover, we have provided the PeDF with the ability to supply personal data and user models to data consumers through a Retriever module. The creation of user models entails the incorporation of different components that extract knowledge from personal data. These components have been grouped together in a main component namely Generator.

Summarizing, the PeDF incorporates seven modules:

1. User Manager. It is a vertical module that allows users to interact with PeDF to sign in, activate the incorporation of new data sources, and check and manage authorizations for access to their personal data and user accounts. It implements an identity management infrastructure and privacy solutions.
2. Registrar. This module allows data consumers to ask for the incorporation of new data sources in order to include new

personal data in the PeDF. It interacts with the User Manager module to obtain the user consent.

3. Collector. This module is in charge of obtaining personal data from external data sources, checking user authorization. It can also include crawlers' components that get personal data from public data sources.
4. Registry. It allows the PeDF to store pointers to external personal data that the PeDF is able to recover from data sources.
5. Generator. It comprises a set of components that allow PeDF to obtain user models from personal data. These implement different techniques of user modelling to uncover user needs, preferences, interests, etc.
6. User Data Store. It is a central repository that stores the personal data that is obtained from external data sources or by the Generator module. It contains different interfaces that allow the updating and refreshing of personal data.
7. Retriever. This module is in charge of communicating with data consumers who are interested in obtaining personal data and user models of a specific user. It interacts with the User Manager module to check user consent and with the Registry or User Data Store to retrieve the personal data requested.

4 FRAMEWORK VALIDATION

We have validated the PeDF in a banking scenario which considers a person-to-person payment service namely PosdataP2P, and the social network Facebook as data sources. Moreover, it includes a financial service called FriendLoans that uses user models from the PeDF to offer its users recommendations about microloans. It is an integration effort to provide user models that fulfill individual business needs of third entities. We have focused our work on a centralized integration based on semantic technologies, which improve the user modelling process. Moreover, we have validated the PeDF with five beta testers from our research group.

Figure 2 represents our validation to the PeDF. Here, we can observe the two private data sources (PosdataP2P and Facebook), the data consumer (MicroLoans), the user and the main PeDF modules that we have validated: User Manager, Collector, User Data Store, Generators, and Retriever.

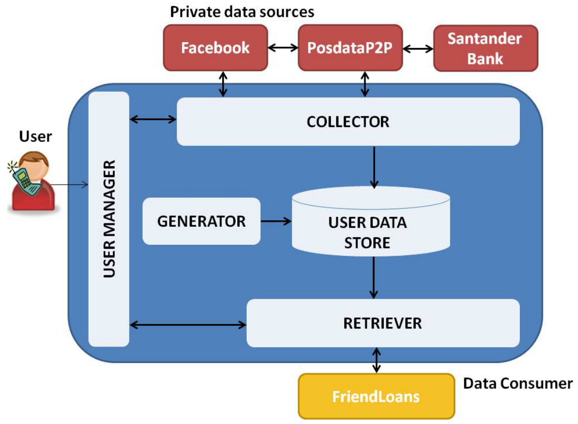

Figure 2. Personal Data Framework validation architecture

4.1 External data sources

We have considered two private data sources for PeDF validation: PosdataP2P service, and the social network Facebook.

PosdataP2P service [17] is an innovative financial service developed within the context of a COM project. It allows Santander University Smart Card (USC) holders to make payments to or request money from friends, using alternative social channels such as texting systems e.g. Telegram, or online social networks e.g. Facebook or Twitter.

The USC is a smart card issued by over 300 universities in collaboration with Santander Bank. It is used by 7.8 million people worldwide to access university services, such as libraries, control access (for example, to computers, campus, sports pavilions, etc.), electronic signature, discounts at retailers, etc. It can be also used to gain access to Santander Bank financial services, working as a credit/debit card linked to the holder's saving account.

To use PosdataP2P service, USC holders have to activate the service first, providing their USC information. Then, they choose the social channels that they want to use to carry out financial transactions. Having done that, students can start making financial transactions by simply posting messages to their friends within their enabled social channels (Figure 3).

Figure 3. PosdataP2P screenshot using Facebook as a channel

The PosdataP2P service generates financial data on USC holders, which is properly recovered by the PeDF in real time. Specifically, the PosdataP2P has an interface to notify financial transaction to PeDF.

The PeDF also obtains demographic and social data from Facebook with user consent. It is based on the Facebook Login and the Facebook Graph API as mentioned in Section 2.

4.2 A Personal Socio-Economic Network

The PeDF validation applies a centralized approach where personal data obtained from external data sources are stored in a central repository. Specifically, it is based on a semantic modelling and storing, and an ontology, namely the Personal Socio-Economic Network (PSEN).

The PSEN represents the exchange of money between people and user social data. We have considered the reusing of existing ontologies, which is a must to allow semantic and syntactic interoperability. Thus, we have identified the FOAF ontology as the best alternative for representing people in a social network context and the SUMO's financial ontology (using the OWL version) for representing the financial concepts. We have also extended them and linked the different socio-economic concepts. The nomenclature that we have used to represent the PSEN concepts is based on SUMO terms so it can be easily related to the upper ontology.

Briefly, the PSEN includes the main terms to describe people, the relationships between them, and the financial data and activities carried out between them (Figure 4). We represent people as the *Person* class from FOAF and we use the corresponding FOAF properties to describe their user's demographic information: *firstName*, *lastName*, *gender*, *age*, *birthday*, and *mbox* (omitted in Figure 4 for the sake of simplicity). We also made use of the *Online Account* class from FOAF that allows the modelling of different web identities or online accounts of a person. We have extended it to include online payment and banking accounts. The former is devoted to service providers that allow users to carry out payment operations through the Internet, such as PosdataP2P service. It has associated a *BankCard* or a *Financial Account* class from the SUMO financial ontology that denotes where the payment will become effective. These classes have a relationship (namely, *cardAccount*) since a *BankCard* is always associated with a *FinancialAccount.* On the other hand, the *Online Banking Account* class represents online banking services including financial institutions, such as Santander Bank.

To model user economic activities, we have defined a *SocialInteraction* class within the PSEN ontology. It includes three main properties: *timestamp*, *channel* and *patient.* The *timestamp* and *channel* properties indicate when and where the social interaction happens respectively, and *patient* designates an *Entity* that participates in the social interaction, i.e. the money exchange. The *SocialInteraction* class also has two subclasses: *Transaction* and *Communication* that have *Payment* and *Request* subclasses correspondingly. These are related to a *hasPayment* link that indicates whether a request for money has been paid.

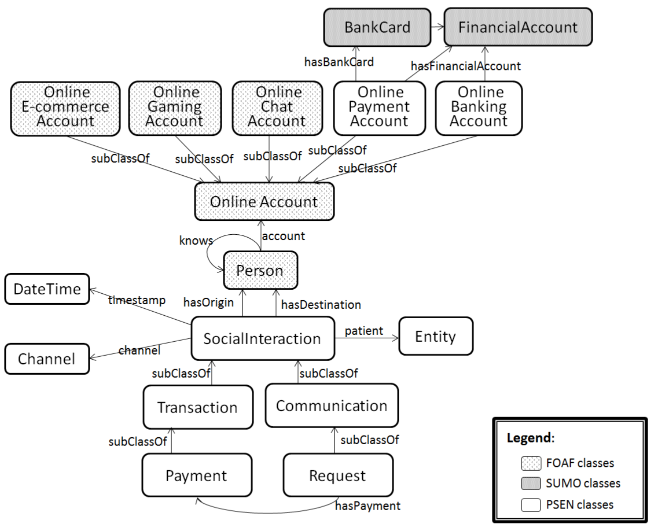

Figure 4. Personal Socio-Economic Network definition

In Figure 4, the rounded rectangles characterize the main concepts and the edges indicate the relationships between two classes. We have distinguished the terms of the different ontologies with darker rectangles indicated in the legend of the figure.

4.3 Knowledge retrieval

We have validated the retrieval of user knowledge through the FriendLoans service, which is based on friendsourcing [31]. It is a form of crowdsourcing where the user's social network is mobilized to achieve a specific objective. Specifically, FriendLoans relies on the PSEN data to offer financial recommendations on microloans to raise money from friends. It has been implemented as a web application in which authenticated users can ask for money from their friends. Basically, a user accesses to the service, indicates the money needed (Figure 5 at the top) and the service provides a list of prospective borrowers who are trusty, available, and solvent enough to lend (Figure 5 at the bottom). Figure 5 shows an example of the FriendLoans service for a user called Maria who needs 200€ from her friends.

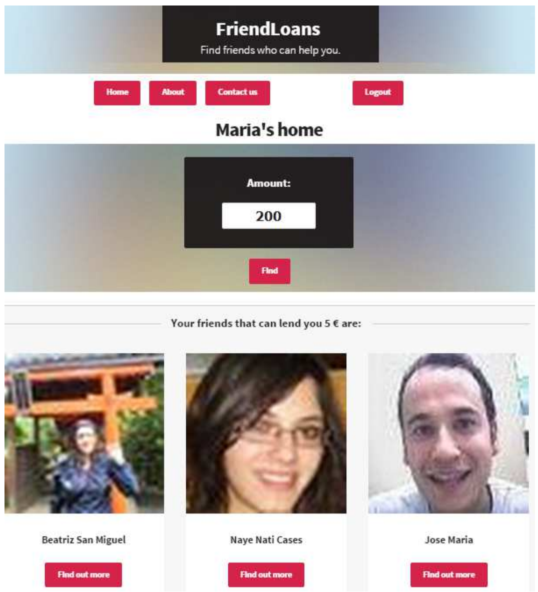

Figure 5. Screenshot of FriendLoans for a user called Maria

Generating a list of friends for a user requires user models that are unknown to FriendLoans, but can be retrieved from the PeDF. The PeDF has incorporated two mechanisms that allow data consumers to ask for user financial relationships and other banking information, all with the consent of the user. Specifically, the PeDF abstracts a set of SPARQL sentences and calls the reasoners which obtain and derive additional knowledge from the PSEN.

The SPARQL sentences obtain personal data and user models directly from the PSEN which can be used by FriendLoans. This information does not derive facts or inferences under the PSEN data, just data contained in it. For example, the list of friends for a specific user, if a person has carried out payments or requests for money, if a person has received money, if a person has requests for money and no associated payments, etc.

As regards the reasoners, they include the mechanisms that allow the extration of derived data. For this, we have implemented four custom rules that detect: 1) whether a user knows another user *A*; 2) whether a user owes money to a user *A*; 3) whether a user has received a payment greater than *X* euros; and 4) whether a user has requests for money with greater amount of money than *Y* euros. In the rules, the user *A* and the amount of money *X* and *Y* can be indicated by FriendLoans to give recommendations to its users. In this way, for the example shown in Figure 5, *A* will be the authenticated user Maria who needs money from her friends, *X* and *Y* could be at least 200€ or the amount wanted by FriendLoans. The results obtained from executing these rules are a set of users that fulfill all conditions. This set is not ordered since the order of execution of the rules is not predictable in the reasoner. However, the PeDF has implemented an algorithm that orders the results including tags that indicate the prioritization.

The next program listing shows an example of a rule that tags the results as the most important ones (it is indicated by the tag *isFirstFor*) for the user Maria (specified by the second line of the rule). The conditions of the rule are: 1) a user who has debts with Maria (defined in a function called *hasDebtWith*), and 2) a user has not requested an amount of money greater than 5€ with other people (defined in a function called *possibleProblem*).

```
[isFirst:
(?Maria psen:isTarget “true”^^xs:Boolean)
(?person psen:hasDebtWith psen:Maria)
noValue(?ecAct           psen:possibleProblem
“true”^^xs:Boolean)
-> (?person psen:isFirstFor ?Maria)]
```

4.4 Identity management and privacy

We have based our identity management infrastructure on OAuth 2.0, as it has become the de facto standard to gain access to personal data on the Web. The User Manager includes the component that manages the interaction with external sites.

Users can currently link their accounts on the PosdataP2P service and Facebook to the PeDF. The process works as follows: when a user activates a data source (i.e. Facebook), he is then redirected to the service provider site to grant the PeDF the required level of authorization. If successful, the data source delivers a token that allows access to the user profile.

As regards privacy, the PeDF has been designed to observe European privacy and data protection principles following a privacy-by-design approach. The User Manager is also the key component here, since it provides users with an identity and privacy dashboard allowing them to 1) grant/revoke consent to the collection, processing and disclosure of their personal data, 2) check the PeDF privacy policies, 3) manage the personal data known and stored by the PeDF, their sources, and the details on the disclosures to third parties as well as exercising their right to access, rectify, erase or block personal data. At the same time, the User Data Store implements security safeguards to avoid and mitigate privacy threats derived from malicious attackers or unwitting users. Finally, as regards the data minimization principle, the use of reasoners allows third parties to be limited and allows justified users to be able to query and retrieve that specified and agreed to by the data subject.

5 RELATED WORK

The PeDF is an ambitious solution that covers four main technological challenges related to personal data: collection, integration, retrieval, and identity and privacy management. These have been widely analyzed separately over time in different contexts, and we can find many researchers addressing each of them in depth. For example, the previously cited literature [10] includes a study into data integration in business environments, or [32] presents the user modelling techniques, its challenges and the state-of-the-art research, focusing on ubiquitous environments.

We can find aligned systems that attempt to solve the same issues as the PeDF in the personal data context. For example, the so-called data brokers [33] are companies that collect personal data on individual (generally, from public data sources), and resell them to or share them with third parties. These systems are focused on data collection and integration, but individuals are generally unaware of their activities. Otherwise, there are a number of companies and projects within the initiative called Personal Cloud[10]. It advocates the creation of safe places where users have complete control of their data. The associated solutions address the definition of a new interaction model between users, service providers, and devices, where clouds connect voluntarily to services which use stored personal data. They focus on identity management, encryption, data storage, cloud computing, as well as other user modelling works related to reputation. Closely related to these, there are different identity management systems [34] that implement end-user solutions with the goal of making personal data available only to the right parties, establishing trust between parties involved, avoiding the abuse of personal data, and making these provisions possible in a scalable, usable, and cost-effective manner. These latter solutions do not generally include user modelling techniques.

On the other hand, there are also specialized systems, namely Generic User Modelling Systems [35] that can serve as a separate user modelling component to different service providers. They address issues related to data representation, inferential capabilities, management of distributed information, or privacy. However, they focus on the reuse of technological user modelling components rather on the reuse of the personal data and user models themselves. Finally, there are solutions referred as Personal Data Store, Personal Data Locker, or Personal Data Vault that roughly describe the same concept. Generally, these solutions are based on a central place where the user can save and manage all their personal data, including data such as text, passwords, images, video or music [36]. These solutions have an end-user approach.

To summarize, the aforementioned solutions are rather diverse from one another, and each of them focuses on a main objective (i.e., personal data collection, identity management, and data storage). Our work is an integration effort to provide an end-to-end solution that aims at incorporating the best solutions for each issue. Our first approach is based on integrating social and financial data. To the best of our knowledge, this is the first effort in this context.

6 CONCLUSIONS AND FUTURE WORK

In this paper we have presented a comprehensive framework intermediating between users and organizations to support the seamless integration of personal data from several, distributed sources and generating advanced knowledge on users, to be shared with interested third parties, all supervised by the users who control and manage the flow of their personal data. The framework includes components for personal data collection, integration, and retrieval, as well as users’ identity and privacy management.

[10] http://personal-clouds.org

The framework has been validated in a financial context, integrating social information from Facebook and a person-to-person payment service, to generate knowledge useful for a personal lending application.

Our future work includes advancing on the design of the privacy-preserving elements required to minimize the personal information retrieved by the data consumers while keeping it useful enough to fit their business needs. These developments will comprise advanced privacy enhancing technologies for attribute-based credentials and database privacy.

ACKNOWLEDGEMENTS

This work is part of the Center for Open Middleware (COM), a joint technology center created by Universidad Politécnica de Madrid, Banco Santander and its technological divisions ISBAN and PRODUBAN.

REFERENCES

[1] I. Barri, T. Loilier, M. van Rijn, A. Stolk, and H. Vasiliadis, 'Open innovation in the financial services sector - Why and how to take action', Technical report, GFT Technologies AG, (2014).

[2] J. P., Moreno, Harvard Business Publishing, *Banks' New Competitors: Starbucks, Google, and Alibaba.* https://hbr.org/2014/02/banks-new-competitors-starbucks-google-and-alibaba/

[3] Open Data Institute and Fingleton Associates, 'Data Sharing and Open Data for Banks', Technical report, (2014).

[4] European Commission, *Protection of personal data.* http://ec.europa.eu/justice/data-protection/

[5] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures,* Ph.D. dissertation, University of California, 2000.

[6] Internet Engineering Task Force (IETF), 'The JavaScript Object Notation (JSON) Data Interchange Format', Proposed Standard RFC 7159, (2014).

[7] W3C, *OpenSocial Foundation Moves Standards Work to W3C Social Web Activity,* http://www.w3.org/blog/2014/12/opensocial-foundation-moves-standards-work-to-w3c-social-web-activity/

[8] G. Gouriten and P. Senellart, 'API Blender: A Uniform Interface to Social Platform APIs', *CoRR,* abs/1301.2086, (2013).

[9] M. Lenzerini, 'Data integration: A theoretical perspective', in *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems,* PODS '02, pp. 233–246, New York, NY, USA, (2002). ACM.

[10] P. Ziegler and K. R. Dittrich, 'Three decades of data integration - all problems solved?', in *18th IFIP World Computer Congress (WCC 2004),* Volume 12, Building the Information Society, pp. 3–12, (2004).

[11] N. Guarino, 'Formal ontology and information systems', in *FOIS98*, pp. 3–15, Trento, Italy, (1998). IOS Press.

[12] M. Uschold and M. Gruninger, 'Ontologies: Principles, methods and applications', *Knowledge Engineering Review*, 11(2), 93–136, (1996).

[13] I. Niles and A. Pease, 'Towards a standard upper ontology', in *Proceedings of the International Conference on Formal Ontology in Information Systems, FOIS01*, pp. 2–9, New York, NY, USA, (2001). ACM.

[14] D. Brickley and L. Milller, 'Foaf vocabulary specification 0.99', Namespace Document - Paddington Edition, (2014).

[15] Object Management Group, *Financial Services Standards.* http://www.omg.org/hot-topics/finance.htm

[16] L. Yu, *A Developer's Guide to the Semantic Web*, Springer, 2011.

[17] B. San Miguel, J. M. del Alamo, J. C. Yelmo, 'Creating and Modelling Personal Socio-Economic Networks in On-Line Banking' in *7th International Workshop on Personalization and Context-Awareness in Cloud and Service Computing, PCS 2014,* pp. 177–190, (2015) Springer [In press].

[18] World Wide Web Consortium (W3C), 'OWL Web Ontology Language', W3C Recommendation, (2004).

[19] N. P. de Koch, *Software engineering for adaptive hypermedia systems: reference model, modeling techniques and development process*, Ph.D. dissertation, Ludwig Maximilians University Munich, 2001.

[20] S. Gauch, M. Speretta, A. Chandramouli and A. Micarelli, 'User profiles for personalized information access', in *The Adaptive Web*, eds., P. Brusilovsky, A. Kobsa, and W. Nejdl, Springer-Verlag, (2007).

[21] P. Brusilovsky, and E. Millán, 'User Models for Adaptive Hypermedia and Adaptive Educational Systems', in *The Adaptive Web*, eds., P. Brusilovsky, A. Kobsa, and W. Nejdl, Springer-Verlag, (2007).

[22] J. Kay, 'Lies, damned lies and stereotypes: Pragmatic approximations of users', in *Proceedings of the Fourth International Conference on User Modeling*, pp. 175–184, Hyannis, MA, (1994). ACM.

[23] M. Montaner, B. López, and J. L. de la Rosa, 'A Taxonomy of Recommender Agents on the Internet', *Artificial Intelligence Review*, 19(4), 285–330, (2003).

[24] S. Sosnovsky, and D. Dicheva, 'Ontological technologies for user modelling', *International Journal of Metadata, Semantics and Ontologies,* 5(1), 32–71, (2010).

[25] World Wide Web Consortium (W3C), *SPARQL Current Status.* http://www.w3.org/standards/techs/sparql#w3c_all

[26] J. M. del Alamo, M. A. Monjas, J. C. Yelmo, B. San Miguel, R. Trapero, and A. M. Fernandez, 'Self-service privacy: User-centric privacy for network-centric identity.', in *Trust Management IV. 4th IFIP WG 11.11 International Conference on Trust Management, IFIPTM 2010,* pp. 17–31, Morioka, Japan, (2010). Springer Berlin Heidelberg.

[27] OASIS, *OASIS Security Services (SAML) TC.* https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[28] O. Manso, M. Christiansen, and G. Mikkelsen, 'Comparative analysis - Web-based identity management systems', Technical report, The Alexandra Institute, (2014).

[29] G. Danezis, J. Domingo-Ferrer, M. Hansen, J. H. Hoepman, D. L. Metayer, R. Tirtea, and S. Schiffner, 'Privacy and Data Protection by Design – from policy to engineering' Technical report, European Union Agency for Network and Information Security (ENISA), (2014).

[30] A. Crespo García, N. Notario McDonnell, C. Troncoso, D. Le Métayer, I. Kroener, D. Wright, J. M. del Álamo and Y. S. Martín, 'D1.2: Privacy and Security-by-design Methodology', Technical report, PRIPARE (2014).

[31] M. S. Bernstein, D. Tan, G. Smith, M. Czerwinski, and E. Horvitz, 'Personalization via friendsourcing', *ACM Transactions on Computer-Human Interaction (TOCHI),* 17(2), 6:1–6:28, (2008).

[32] J. Kay T. Kuflik, and B. Kummerfeld, 'Challenges and solutions of ubiquitous user modeling', in *Ubiquitous Display Environments,* eds., A. Krüger and T. Kuflik, Springer Berlin Heidelberg, (2012).

[33] E. Ramirez, J. Brill, M. K. Ohlhausen, J. D. Wright, T. McSweeny, 'Data Brokers – A Call for Transparency and Accountability', Technical report, Federal Trade Commission, (2014).

[34] E.Bertino and K. Takahashi, *Identity Management: Concepts, Technologies, and Systems*, Artech House, Inc., 2010.

[35] A. Kobsa, 'Generic user modeling systems', *User Modeling and User-Adapted Interaction,* 11(1-2), 49–63, (2001).

[36] M. Sabadello, 'Startup Technology Report – Phase One: Acquiring, Storing, Accesing and Managing Personal Data', Technical report, Personal Data Ecosystem Consortium, (2014).

Human Computation Based Acquisition Of Financial Service Advisory Practices

Alexander Felfernig[1] and **Michael Jeran**[1] and **Martin Stettinger**[1] and **Thomas Absenger**[1] and **Thomas Gruber**[1] and **Sarah Haas**[1] and **Emanuel Kirchengast**[1] and **Michael Schwarz**[1] and **Lukas Skofitsch**[1] and **Thomas Ulz**[1]

Abstract. Knowledge-based recommenders support an easier comprehension of complex item assortments (e.g., financial services and electronic equipment). In this paper we show (1) how such recommenders can be developed in a Human Computation based knowledge acquisition environment (PEOPLEVIEWS) and (2) how the resulting recommendation knowledge can be exploited in a competition-based e-Learning environment (STUDYBATTLE).

1 Introduction

Knowledge-based recommenders [2] support users on the basis of semantic knowledge about the item (product) domain.[2] One variant of knowledge-based recommenders are *constraint-based recommenders* [8] which exploit explicit constraints (rules) that encode the recommendation knowledge. Another variant are *critiquing-based recommenders* [4]: new items are presented to the user as long as the user is unsatisfied and articulates critiques (e.g., an item should be *cheaper*). In critiquing-based recommendation, new items are determined by similarity functions. For a detailed overview of recommendation approaches we refer to [3, 20].

In this paper we focus on constraint-based recommenders, i.e., recommenders that are based on explicit recommendation rules (constraints). The development of such recommenders is often a time-consuming and error-prone process which can be primarily explained by the *knowledge acquisition bottleneck*: in the formalization of product domain and recommendation knowledge, misunderstandings can occur and as a result knowledge engineers encode this knowledge in an unintended fashion. The more recommenders have to be developed and maintained the higher the risk that the organization runs into a scalability problem where additional resources are needed to be able to perform knowledge engineering and maintenance.

An alternative to the hiring of additional staff for development and maintenance of recommendation knowledge bases is to change the underlying knowledge engineering paradigm. The idea of PEOPLEVIEWS is to engage domain experts more deeply into knowledge engineering tasks. We do not want to "convert" them into technical experts but to define basic tasks (*micro tasks*) that are easy to understand and complete even for domain experts without the corresponding technical expertise. Micro tasks completed by users provide knowledge chunks that can be aggregated into a PEOPLEVIEWS recommender knowledge base.

The resulting PEOPLEVIEWS recommenders support customers (and especially in the financial services domain also sales representatives) in finding products that fit their wishes and needs. Using such a recommender, items are retrieved within the scope of a dialog (these systems are often also denoted as conversational) where users articulate their requirements and the system tries to identify corresponding solutions. Major advantages of such systems are reduced error rates in the phase of order acquisition, more time that can be invested in contacting new customers due to fewer errors, more satisfied customers, and also pre-informed customers due to the fact that recommender applications can be made publicly available.

Knowledge-based recommender systems have been applied in various item domains – due to the diversity of applications, we can only give some examples of applications of these systems. In the financial services domain, for example, the following applications of knowledge-based recommendation technologies are reported in the literature. Felfernig et al. [11, 12] show an application in the context of investment decisions where recommenders are provided to sales representatives who exploit the recommenders in sales dialogs. Time savings are reported as one of the major improvements directly related to the application of recommendation technologies. Another application of knowledge-based technologies in financial services is presented by Fano and Kurth [7] who introduce a simulation environment that can directly visualize the effects of financial decisions on the financial situation of a family.

Felfernig et al. [9] present a digital camera recommender deployed on a large Austrian product comparison platform. Peischl et al. [22] show the application of constraint-based recommendation technologies in the domain of software effort estimation. WEEVIS[25][3] is a MediaWiki[4] based environment for the development and maintenance of constraint-based recommender applications – a couple of freely available recommenders have already been deployed. Knowledge-based technologies for the recommendation of business plans are introduced by Jannach and Bundgaard-Joergensen [19]. The recommendation of equipment configuration in the context of smarthomes is introduced by Leitner et al. [21]. Technologies that recommend changes in software development practices are introduced by Pribik and Felfernig [23]. Finally, Burke and Ramezani [5] show how to select recommendation algorithms by introducing rules for *recommending recommenders*.

[1] Applied Software Engineering, Institute for Software Technology, Graz University of Technology, Austria, email: {felfernig, mjeran, stettinger}@ist.tugraz.at, {thomas.absenger, th.gruber, sarah.haas, emanuel.kirchengast, michael.schwarz, lukas.skofitsch, thomas.ulz}@student.tugraz.at.

[2] The terms *item* and *product* are used synonymously throughout the paper.

[3] www.weevis.org.

[4] www.mediawiki.org.

In PEOPLEVIEWS, principles of Human Computation [26] are included into the development of knowledge-based recommenders. The idea of Human Computation is to let persons perform tasks in which they are better than computers, for example, the identification of product properties from a website. In the context of knowledge base development and maintenance the idea is to let domain experts perform tasks they are much better in compared to knowledge engineers who typically have less knowledge about the product domain and thus relieve the work of knowledge engineers. MATCHIN [18] is based on the idea of preference elicitation by asking users what a person would typically prefer when having to choose between alternatives. Compared to this work, PEOPLEVIEWS allows to derive constraint-based recommenders which are the basis for intelligent user interfaces that support, for example, deep explanations [17] and the diagnosis and repair of inconsistent requirements [13, 14].

The major contributions of this paper are the following. First, we show how financial service recommender knowledge bases can be developed by a community of domain experts. Second, we sketch how such knowledge bases can also be exploited for teaching advisory practices on the basis of games (STUDYBATTLE environment). Third, we provide a discussion of major issues for future research.

The remainder of this paper is organized as follows. In Section 2 we introduce basic concepts of Human Computation based knowledge construction. To give an impression of the PEOPLEVIEWS and the STUDYBATTLE user interface, we present example screenshots in Section 3. Preliminary results of empirical evaluations are shortly discussed in Section 4. In Section 5 we provide an overview of issues for future work. We conclude the paper with Section 6.

2 Developing PEOPLEVIEWS Recommenders

The PEOPLEVIEWS environment supports two basic modes of interaction. First, recommender applications can be created in the *modeling mode* and second, the applications can be executed in the *recommendation mode*. In this section we discuss different tasks to be performed in order to create a PEOPLEVIEWS recommender. Table 1 provides an overview of the users of our working example. These users will jointly develop a PEOPLEVIEWS recommender.

user	email	pwd
Andrea	andrea@...	****
Mary	mary@...	*****
Luc	luc@...	******
Torsten	torsten@...	****

Table 1. Example users of PEOPLEVIEWS environment.

Table 2 contains an overview of items (financial services) that are used in our working example. The *Investment Funds* (A and B) have a higher risk of loss and require that customers have a high willingness to take risks, otherwise these services will not be recommended. *Building Loan*, *Bond*, and *Savings Book* are lower-risk items. In the current version of PEOPLEVIEWS, items can be characterized by additional item attributes, however, these attributes are not used by recommendation rules constructed from micro contributions.

In PEOPLEVIEWS, user requirements $req_i \in REQ$ are specified as assignments of *user attributes*. For our financial services recommender we define a set of user attributes which are enumerated in Table 3. In the current version of the system, user attributes are defined by the creators of a recommender application, i.e., attribute definitions can not be extended by other users who contribute to the further development of the application on the basis of micro tasks.

id	item name
Φ_1	Investment Fund A
Φ_2	Investment Fund B
Φ_3	Building Loan
Φ_4	Bond
Φ_5	Savings Book

Table 2. Example set of items used in working example.

user attribute	question to user	attribute domain
goal (gl)	What are your personal goals?	{Studies, Pension, Speculation, Car, House, World trip, noval}
runtime (rt)	When is the money needed?	{in 1 year, in 2 years, in 3-5 years, in 5-10 years, in 10-20 years, in more than 20 years, noval}
risk (ri)	Preparedness to take risks?	{low, medium, high, noval}

Table 3. User attributes $u \in U$ of example financial services recommender.

In the PEOPLEVIEWS *recommendation mode*, user attributes can be used to specify user (customer) requirements $req_i \in REQ$. In the *modeling mode*, user attributes represent a central element of a micro task: given a certain item, users are asked to estimate which values of user attributes are compatible with the item, i.e., are a criteria for selecting and recommending the item. The evaluation of items with regard to user attributes is the central micro task implemented in the current PEOPLEVIEWS prototype. A detailed evaluation of the example items (Table 2) regarding the user attributes *goal*, *runtime*, and *risk* is provided in Table 4.

Each row of Table 4 specifies a so-called *user-specific filter constraint* [10], i.e., a filter constraint (specified by a user) regarding a specific item. For example, user *Luc* specified *Pension* and *Speculation* as possible goals that lead to an inclusion of the item *Investment Fund B* into a recommendation. Furthermore, *Luc* believes that a user should have a high preparedness to take risks (attribute *risk*) and should need the payment in 3-5 years, 5-10 years or 10-20 years from now on. Semantically, an item X is selected by a user-specific filter constraint if all the preconditions are fulfilled.

In order to derive *recommendation-relevant filter constraints* (recommendation rules) [10]), user-specific filter constraints have to be aggregated. An example of this aggregation step is depicted in Table 5. For each item all related user-specific filter constraints are integrated into one constraint. Each row in this table has to be interpreted as a filter constraint for a specific item, for example, the constraint in the first row of Table 5 is the following. The item Φ_1 (*Investment Fund A*) is *included* (recommended) if the user requirements regarding goal (gl), runtime (rt), and risk (ri) are consistent with the condition of the recommendation-relevant filter constraint $gl \in \{$*Studies, Pension, Speculation, noval*$\} \wedge rt \in \{$*in 5-10 year, in 10-20 years, noval*$\} \wedge ri \in \{$*medium, high, noval*$\} \rightarrow$ *include*(Φ_1).

Table 5 includes the complete set of recommendation-relevant filter constraints (recommendation rules). Exactly these conditions are applied by PEOPLEVIEWS to determine recommendations for a user. In PEOPLEVIEWS, each item has exactly one related recommendation-relevant filter constraint; each such filter constraint is represented by one row in Table 5. The general logical representation of a recommendation-relevant filter constraint f for an item Φ is shown in Formula 1. In this context, $values(\Phi, u)$ is the set of

user	item name (id)	goal	runtime	risk
Andrea	Investment Fund A (Φ_1)	Studies, Pension, Speculation	in 5-10 years, in 10-20 years	high
Luc	Investment Fund A (Φ_1)	Pension, Speculation	in 5-10 years, in 10-20 years	high
Mary	Investment Fund A (Φ_1)	Pension, Speculation	in 5-10 years, in 10-20 years	medium, high
Torsten	Investment Fund B (Φ_2)	Pension, Speculation	in 3-5 years, in 5-10 years, in 10-20 years	high
Luc	Investment Fund B (Φ_2)	Pension, Speculation	in 3-5 years, in 5-10 years, in 10-20 years	high
Mary	Building Loan (Φ_3)	Studies, Pension, Car, House	in 5-10 years, in 10-20 years	low, medium, high
Andrea	Building Loan (Φ_3)	Studies, Pension, Car, House	in 5-10 years	low, medium
Luc	Building Loan (Φ_3)	Studies, Pension, Car, House	in 5-10 years	low, medium
Mary	Bond (Φ_4)	Studies, Car, House	in 2 years, in 3-5 years, in 5-10 years	low, medium
Andrea	Savings Book (Φ_5)	Studies, Car, House, World trip	in 1 year, in 2 years, in 3-5 years, in 5-10 years	low
Torsten	Savings Book (Φ_5)	Studies, House, World trip	in 1 year, in 2 years, in 3-5 years, in 5-10 years	low

Table 4. Example of *user-specific filter constraints* (= micro contributions).

supported domain values of user attribute $u \in U$ (see Table 4). The constant $noval$ denotes the fact that no value has been selected for the corresponding user attribute.

$$f(\Phi) : \bigwedge_{u \in U} u \in values(\Phi, u) \cup \{noval\} \rightarrow \text{include}(\Phi) \quad (1)$$

For each pair $(\Phi, val \in values(\Phi, u))$, PEOPLEVIEWS determines a corresponding support value (see Formula 2). In this context, $occurrence(\Phi, val)$ denotes the number of times, value val occurs in a user-specific filter constraint for item Φ and $occurrence(\Phi)$ denotes the number of times an item Φ is referred in a user-specific filter constraint. For example, $support(\Phi_1, \text{Studies}) = \frac{1}{3}$.

$$support(\Phi, val) = \frac{occurrence(\Phi, val)}{occurrence(\Phi)} \quad (2)$$

The complete set of support values is depicted in Table 6. In PEOPLEVIEWS, an item Φ can have an associated *rating* ($rating(\Phi)$) which represents an item evaluation with regard to quality and related services. Such a rating can be determined, for example, by calculating the average of the individual user item ratings.[5] For simplicity, we do not take into account user ratings in the utility function discussed below (see Formula 3).

Depending on the requirements articulated by the current user (see, e.g., Table 7), PEOPLEVIEWS determines and ranks a set of relevant items as follows. First, recommendation-relevant filter constraints are applied to pre-select items that fulfill the user requirements $REQ = \{req_1, req_2, ..., req_k\}$. In our example, the set $\{$*Investment Fund A, Building Loan*$\}$ would be selected by the recommendation-relevant filter constraints (see Table 5).

[5] Similar to ratings provided by platforms such as *amazon.com*.

item name (id)	attribute:value	support value
Investment Fund A (Φ_1)	goal: Studies	0.33
	goal: Pension, Speculation	1.0
	runtime: in 5-10 years, in 10-20 years	1.0
	risk: medium	0.33
	risk: high	1.0
Investment Fund B (Φ_2)	goal: Pension, Speculation	1.0
	runtime: in 3-5 years, in 5-10 years, in 10-20 years	1.0
	risk:high	1.0
Building Loan (Φ_3)	goal: Studies, Pension, Car, House	1.0
	runtime:in 5-10 years	1.0
	runtime:in 10-20 years	0.33
	risk:low, medium	1.0
	risk:high	0.33
Bond (Φ_4)	goal: Studies, Car, House	1.0
	runtime:in 2 years, in 3-5 years, in 5-10 years	1.0
	risk:low, medium	1.0
Savings Book (Φ_5)	goal: Studies, House, World trip	1.0
	goal:Car	0.5
	runtime:in 1 year, in 2 years, in 3-5 years, in 5-10 years	1.0
	risk:low	1.0

Table 6. Support values (see Formula 2) derived from user-specific filter constraints (see Table 4).

item name (id)	goal	runtime	risk
Investment Fund A (Φ_1)	Studies, Pension, Speculation	in 5-10 years, in 10-20 years	medium, high
Investment Fund B (Φ_2)	Pension, Speculation	in 3-5 years, in 5-10 years, in 10-20 years	high
Building Loan (Φ_3)	Studies, Pension, Car, House	in 5-10 years, in 10-20 years	low, medium, high
Bond (Φ_4)	Studies, Car, House	in 2 years, in 2-5 years, in 5-10 years	low, medium
Savings Book (Φ_5)	Studies, Car, House, World trip	in 1 year, in 2 years, in 3-5 years, in 5-10 years	low

Table 5. Example of *recommendation-relevant filter constraints* which are the result of integrating user-specific filter constraints (see Table 4).

id	requirement
req_1	$goal =$ Studies
req_2	$goal =$ Pension
req_3	$runtime =$ in 5-10 years
req_4	$risk =$ medium

Table 7. Example set of user requirements ($req_i \in REQ$).

The determined recommendation set must be ranked before being presented to the user. In PEOPLEVIEWS, item ranking is based on the following utility function (see Formula 3). The utility of each item is derived from the support values of individual requirements (see Formula 2).

$$utility(\Phi, REQ) = \Sigma_{req \in REQ}\ support(\Phi, req) \quad (3)$$

The item ranking of our working example as a result of applying Formula 3 is depicted in Table 8. For example, $utility(\Phi_3, REQ$ = {goal = Studies, goal = Pension, runtime = in 5-10 years, risk = medium}) = $support(\Phi_3$, goal = Studies) + $support(\Phi_3$, goal = Pension) + $support(\Phi_3$, runtime = in 5-10 years) + $support(\Phi_3$, risk = medium) = 1.0 + 1.0 + 1.0 + 1.0 = 4.0.

item name (id)	utility	rank
Building Loan (Φ_3)	4.0	1
Investment Fund A (Φ_1)	2.66	2

Table 8. Utility-based ranking of items in the recommendation set.

3 User Interface

3.1 PEOPLEVIEWS

In this section we discuss the PEOPLEVIEWS user interface[6] and also show how PEOPLEVIEWS recommendation knowledge can be exploited by the STUDYBATTLE learning environment. The PEOPLEVIEWS homescreen is depicted in Figure 1. For applying PEOPLEVIEWS recommenders, there is no explicit need for being logged in. Recommenders can be selected and activated directly from the homescreen (see the tag cloud in Figure 1).

If users are logged in, they are allowed to contribute to the development of PEOPLEVIEWS recommender applications. Only the creators of a recommender application are allowed to define user attributes. Other users can complete micro tasks in terms of evaluating items with regard to a defined set of user attributes. The list of user attributes used in our working example is depicted in Figure 2 (corresponds to the entries of Table 3).

Figure 1. PEOPLEVIEWS homescreen – the current version of the user interface is provided in German. The homescreen explains the basic functionalities of the system (development, maintenance, and execution of recommender applications).

Logged-in users are also allowed to enter new items to the recommender product catalog. The PEOPLEVIEWS representation of product catalogs is exemplified in Figure 3 (corresponds to the list of items shown in Table 2).

The interface for evaluating an item with regard to a set of user attributes is depicted in Figure 4. The screenshot depicts the evaluation of *Building Loan* with regard to the user attribute *goal*. After having completed the definition of a PEOPLEVIEWS recommender,

[6] The user interface is currently only available in German.

Figure 2. PEOPLEVIEWS: example user attributes.

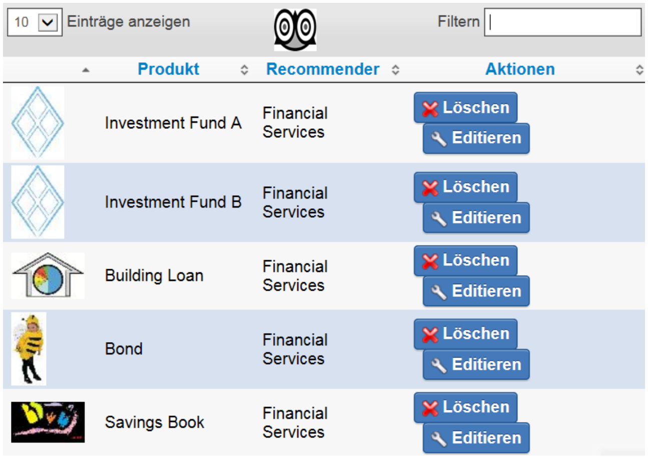

Figure 3. PEOPLEVIEWS: example of an item list.

the recommender can directly be executed. The user interface of our financial services recommender is depicted in Figure 5.

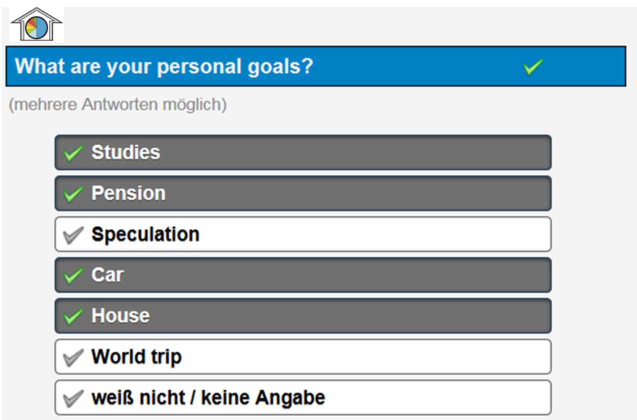

Figure 4. PEOPLEVIEWS: example of an item evaluation user interface (evaluation of item *Building Loan* with regard to the user attribute *goal*).

3.2 STUDYBATTLE

Recommendation-relevant filter constraints can be further exploited for generating different learning applications that are part of the STUDYBATTLE environment. STUDYBATTLE is a game-based learning environment which can be utilized as an environment for learning product knowledge and sales practices. Examples of STUDYBATTLE games are the following.

Assign Properties. Figure 6 depicts an example user interface of a STUDYBATTLE application that implements a quiz related to knowledge about the relationship between user attributes and items. In the example, users have the task to assign items on the left hand side to user attribute values on the right hand side where each product has to be assigned to at least one attribute value and vice-versa.

Find Items. A different version of the game depicted in Figure 6 is to ask for products that fulfill certain criteria (represented by a combination of user attribute settings).

Find Incompatibilities. This game focuses on combinations of user attribute values that do not lead to a solution, i.e., users have to specify combinations of user attribute values from which they think that no corresponding solution could be found.

Maximize Requirements. The task is to identify minimal sets of requirements (from a given set of requirements REQ) that have to be deleted from REQ such that the remaining requirements lead to at least one solution. This game type reflects the principles of model-based diagnosis [6, 24], i.e., support users in learning and improving repair behavior in situations where no solution can be identified.

Maximize Items. A similar task is focused on the repair of item sets; in this context the task of users is to identify a maximal set of items from a given set of items such that there exists at least one combination of user attribute values that lead to these items (not necessarily exclusively). An additional criteria could be that at least n items from the original item list must remain in the result set.

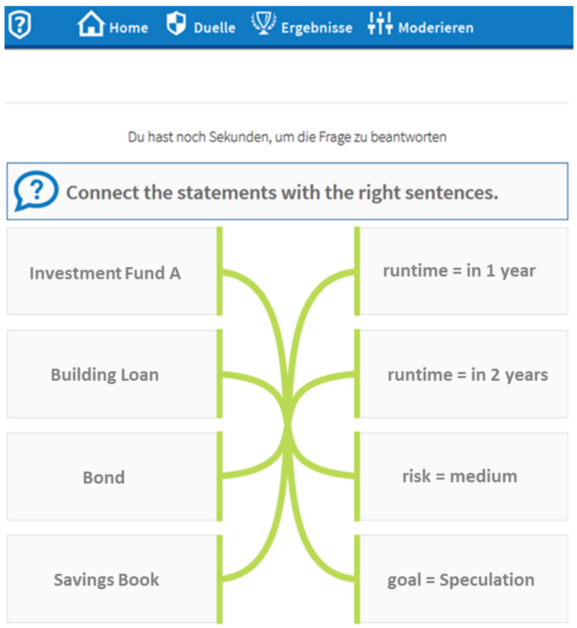

Figure 6. STUDYBATTLE "Assign Properties" learning application. The task of the user is to relate items with corresponding attribute values.

4 Preliminary Evaluation Results

Human Computation based Knowledge Acquisition. Applying Human Computation concepts [26] in the context of recommender application development and maintenance has the potential to lift the burden of enormous engineering and maintenance efforts from the

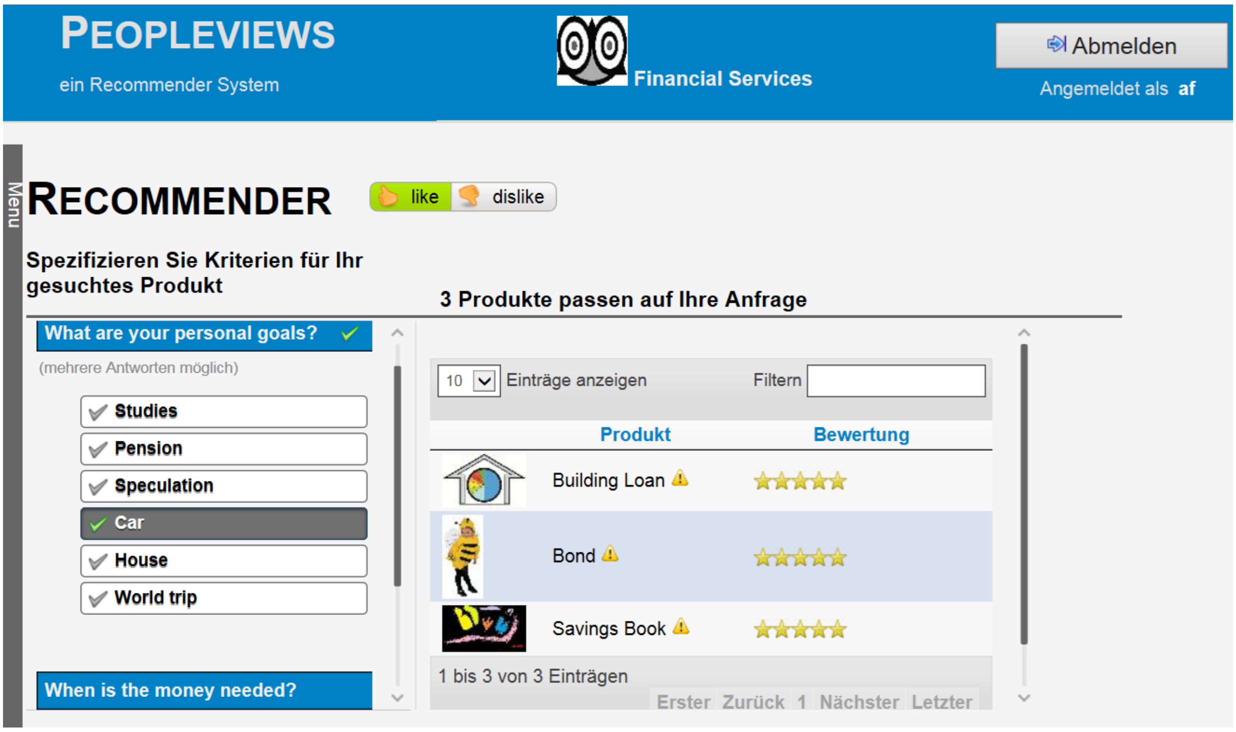

Figure 5. PEOPLEVIEWS: example of a recommender application (Financial Services).

shoulder of knowledge engineers. Micro tasks as sketched in this paper can be structured in a way that they are understandable for domain experts without a computer science background. Knowledge gained from completed micro tasks can be easily integrated into a corresponding recommender knowledge base. Due to the increasing size and complexity of knowledge bases, the development of such technologies is crucial since they help to tackle scalability issues which otherwise could cause a complete failure with regard to a company-wide recommender deployment. As such, PEOPLEVIEWS technologies can be considered as a first step towards more scalable development methods that will also help to further increase the popularity of knowledge-based (recommendation) technologies.

Usability. An initial user study has been conducted with an early version of PEOPLEVIEWS at the Graz University of Technology [10]. N=161 (15% female and 85% male) students interacted with the system with the goal to develop different recommender applications. After having completed the development, the study participants had to complete a questionnaire which was based on the system usability scale (SUS) [1]. Evaluation results regarding the SUS aspects are summarized in Figure 7. Besides usability questions, further feedback has been provided by the study participants, for example, the majority of the participants (69% of all study participants) would like to further contribute to PEOPLEVIEWS recommenders. 56% out of those participants who wanted to contribute agreed to contribute within a time frame of less than 30 minutes per week.

5 Future Work

The major goal of this paper was to provide an overview of the PEOPLEVIEWS recommendation environment. There are many issues for future work that we want to tackle and integrate corresponding solutions in upcoming PEOPLEVIEWS versions.

Weighting of Item Evaluations. In the current PEOPLEVIEWS version it is possible to assign user attribute values to items, i.e., to specify which criteria are relevant for the selection of a certain item. In future versions of PEOPLEVIEWS it will be possible to integrate weights into item evaluations. This maybe does not play a major role in financial service related recommender applications but can be important in other domains were nuances and personal tastes play a more important role. For example, in the context of recommending digital cameras, it can be important to specify degrees regarding certain camera properties, for example, the degree to which a camera is able to support sports photography.

Further Micro Tasks. In the current system version, the only micro task to be completed is to define the relationship (compatibility properties) between items and corresponding user attribute values. In future versions of PEOPLEVIEWS we will extend this list of micro tasks (see Table 9).

User Selection for Micro Tasks. An important enhancement will be the inclusion of methods that automatically select users for a given set of micro tasks and also take into account fairness in the distribution of micro tasks. As detected in our initial studies, users are willing to contribute to the further development of PEOPLEVIEWS recommenders. An important issue in this context is to find the users with the right expertise for certain tasks and also to not overload users. Our approach in this context will be to maintain user profiles which are derived from observing the activities of a user within PEOPLEVIEWS. For example, if a user selects a certain item when interacting with the financial services recommender, the keywords extracted from the corresponding item description are stored in the user profile. If (in the future) micro tasks related to similar items (items with a similar description) have to be completed, users with expertise regarding such items will be the preferred contact persons.

Games. Games will be another mechanism for data collection in

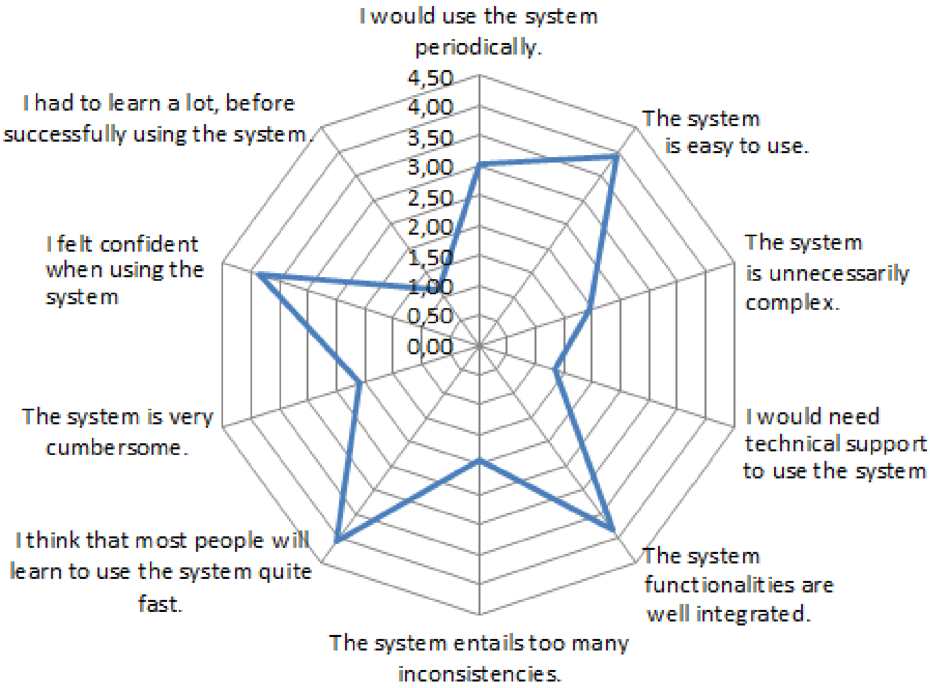

Figure 7. Results of a SUS-based usability study [1] of the PEOPLEVIEWS environment.

name	description
item quality check	check whether a certain item belongs to a specific recommender (is an existing recommender-related item)
attribute quality check	check whether a certain attribute belongs to to a specific recommender (user attribute or item attribute exists in the item domain)
attribute value quality check	check whether a certain value belongs to the domain of an attribute (user attribute or item attribute)
graphic check	check whether a certain figure belongs to a certain item
evaluate item	assign user attribute values to items
attribute value utility check	derive a ranking that shows which items best support a user attribute value

Table 9. Example list of micro tasks to be integrated in PEOPLEVIEWS.

the PEOPLEVIEWS modeling mode. A single user game will be included that is quiz-based. The overall goal is to guess user attribute settings correctly that best describe a certain item. In a second game two users will jointly try to figure out user attribute values that best describe shown items. The more matching item evaluations exist the better the team performs.

Dependencies between User Attributes and Item Attributes. An extension of the current PEOPLEVIEWS version will be the possibility to identify direct relationships between user attribute values and technical product properties. This is not the case in the current PEOPLEVIEWS version since dependencies are only defined between user attribute values and items.

Recommendation Algorithms. The current version of PEOPLEVIEWS relies on the discussed recommendation-relevant filter constraints – item ranking is based on a utility-based evaluation (see Formula 3). In future versions of PEOPLEVIEWS we will extend the quality of recommendation algorithms by, for example, adapting the determination of support values. If, for example, additional information about the performance of a certain user is available (e.g., performance with regard to correctly completed micro tasks in the past), this information can be used to increase/decrease the weight of a user when determining support values. Finally, when users are specifying their requirements, future versions of PEOPLEVIEWS will allow the specification of preferences (weights) which indicate user preferences regarding certain requirements. This will also include approaches to the learning of weights (users should not have to specify all weights explicitly).

Inconsistency Management. Given a set of customer requirements it could be the case that no solution can be presented to the user. In upcoming versions of PEOPLEVIEWS we will focus on integrating state-of-the-art diagnosis algorithms that help to automatically determine repair actions in such inconsistent situations [15]. These repairs will take into account user weights (preferences) and thus minimize the number of interaction cycles needed to find a reasonable solution. In addition to this more intelligent management of inconsistent requirements, we will integrate mechanisms that help to consolidate the set of user-specific filter constraints in order to make the resulting recommendation-relevant filter constraints more compact. Consolidation will be achieved, for example, on the basis of redundancy detection algorithms [16].

Quality Management. The major task of quality management is to assure the quality of the dataset collected on the basis of different micro tasks. Quality assurance must be capable of detecting and preventing manipulations of the dataset (also under the assumption that anonymous users are allowed to complete micro tasks), it must also identify changes to the given set of user-specific filter constraints that help to improve the prediction quality of recommendation algorithms. Quality assurance is also responsible for the generation of micro tasks that need to be completed in order to improve the overall quality of the PEOPLEVIEWS datasets. The micro tasks generated by quality assurance are summarized as an *agenda* – this agenda is forwarded to micro task scheduling that is responsible for distributing micro tasks to the PEOPLEVIEWS user community.

6 Conclusions

In this paper we gave an overview of the PEOPLEVIEWS recommendation environment which exploits concepts of Human Computation to integrate domain experts more deeply into knowledge base development and maintenance processes. PEOPLEVIEWS knowledge bases can be exploited to generate learning applications which can be used in the STUDYBATTLE environment. A major focus of this paper was to show how PEOPLEVIEWS can be applied in the context of financial service recommendation. The concepts presented in this paper have the potential to avoid scalability issues which already exist in many knowledge-based environments due to the increasing size and complexity of knowledge bases.

REFERENCES

[1] A. Bangor, P. Kortum, and J. Miller, 'An Empirical Evaluation of the System Usability Scale (SUS)', *International Journal of Human-Computer Interaction*, **24**(6), 574–594, (2008).

[2] R. Burke, 'Knowledge-based recommender systems', *Encyclopedia of Library and Information Systems*, **69**(32), 180–200, (2000).

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

[4] R. Burke and K. Hammond, 'The FindMe Approach to Assisted Browsing', *IEEE Expert*, 32–40, (1997).

[5] R. Burke and M. Ramezani, 'Matching recommendation technologies and domains', in *Recommender Systems Handbook*, 367–386, Springer, (2010).

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

[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 and R. Burke, 'Constraint-based recommender systems: Technologies and research issues', in *IEEE ICEC'08*, pp. 17–26, Innsbruck, Austria, (2008).

[9] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, 'An environment for the development of knowledge-based recommender applications', *International Journal of Electronic Commerce (IJEC)*, **11**(2), 11–34, (2006).

[10] A. Felfernig, S. Haas, G. Ninaus, M. Schwarz, T. Ulz, M. Stettinger, K. Isak, M. Jeran, and S. Reiterer, 'RecTurk: Constraint-based Recommendation based on Human Computation', in *RecSys 2014 CrowdRec Workshop*, pp. 1–6, Foster City, CA, USA, (2014).

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

[12] 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).

[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] A. Felfernig, C. Zehentner, and P. Blazek, 'Corediag: Eliminating redundancy in constraint sets'.

[17] G. Friedrich, 'Elimination of spurious explanations', in *European Conference on Artificial Intelligence (ECAI 2004)*, pp. 813–817, Valencia, Spain, (2004).

[18] S. Hacker and L. VonAhn, 'Matchin: Eliciting User Preferences with an Online Game', in *CHI'09*, pp. 1207–1216, (2009).

[19] D. Jannach and U. Bundgaard-Joergensen, 'SAT: A Web-Based Interactive Advisor for Investor-Ready Business Plans', in *Intl. Conference on e-Business (ICE-B 2007)*, pp. 99–106, (2007).

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

[21] G. Leitner, A. Fercher, A. Felfernig, and M. Hitz, 'Reducing the Entry Threshold of AAL Systems: Preliminary Results from Casa Vecchia', in *13th Intl. Conference on Computers Helping People with Special Needs*, pp. 709–715, (2012).

[22] B. Peischl, M. Zanker, M. Nica, and W. Schmid, 'Constraint-based Recommendation for Software Project Effort Estimation', *Journal of Emerging Technologies in Web Intelligence*, **2**(4), 282–290, (2010).

[23] I. Pribik and A. Felfernig, 'Towards Persuasive Technology for Software Development Environments: An Empirical Study', in *Persuasive Technology Conference (Persuasive 2012)*, pp. 227–238, (2012).

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

[25] S. Reiterer, A. Felfernig, P. Blazek, G. Leitner, F. Reinfrank, and G. Ninaus, 'WeeVis', in *Knowledge-based Configuration – From Research to Business Cases*, eds., A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, chapter 25, 365–376, Morgan Kaufmann Publishers, (2013).

[26] L. VonAhn, 'Human Computation', in *Technical Report CM-CS-05-193*, (2005).

Case-based Recommender Systems for Personalized Finance Advisory

Cataldo Musto[1] and **Giovanni Semeraro**[1]

1 Abstract

Wealth Management is a business model operated by banks and brokers, that offers a broad range of investment services to individual clients to help them reach their investment objectives. Wealth management services include investment advisory, subscription of mandates, sales of financial products, collection of investment orders by clients. Due to the complexity of the tasks, which largely require a deep knowledge of the financial domain, a trend in the area is the exploitation of recommendation technologies to support financial advisors and to improve the effectiveness of the process.

The talk presents a framework to support financial advisors in the task of providing clients with personalized investment strategies. The methodology is based on the exploitation of case-based reasoning and the introduction of a diversification technique. A prototype of the framework has been used to generate personalized portfolios, and its performance, evaluated against 1,172 real users, shows that the yield obtained by recommended portfolios overcomes that of portfolios proposed by human advisors in most experimental settings.

2 Introduction

Wealth management services have become a priority for most financial services companies. As investors are pressing wealth managers to justify their value proposition, turbulences in financial markets reinforce the need to improve the advisory offering with more customized and sophisticated services. As a consequence, a recent trend in wealth management is to improve the advisory process by exploiting recommendation technologies. However, some peculiarities of the financial domain make hard to put into practice the most common recommendation approaches, as the Content-Based (CB) or the Collaborative Filtering (CF). As regards CB recommenders, the available content, which is necessary to feed a CB recommendation algorithm, is very inadequate and not meaningful, since each user can be just modeled through her *risk profile*[2] along with some demographical features. Similarly, financial products are described through a *rating*[3] provided by credit rating agencies, an average *yield* on different time intervals and the *category* it belongs to. In this recommendation setting a pure CB strategy is likely to fail, since the overlap between features is very poor. Moreover, the over-specialization problem [1], typical of CB recommenders, may collide with the fact that turbulence and fluctuations in financial markets suggest to change and diversify the investments over time. Similarly, CF algorithms can hardly be adopted because of the well-known *sparsity* problem, which makes very difficult to identify the neighbors of the target user.

These dynamics suggest to focus on different recommendation paradigms. Given that financial advisors have to analyze and sift through several *investment portfolios*[4] before providing the user with a solution able to meet her investment goals, the insight behind our recommendation framework is to exploit Case-Based Reasoning (CBR) to tailor investment proposals on the ground of a case base of previously proposed investments.

3 Methodology

Our recommendation process is based on the typical CBR workflow described in [2] and sketcted in Figure 3. Our pipeline is structured in three different steps:

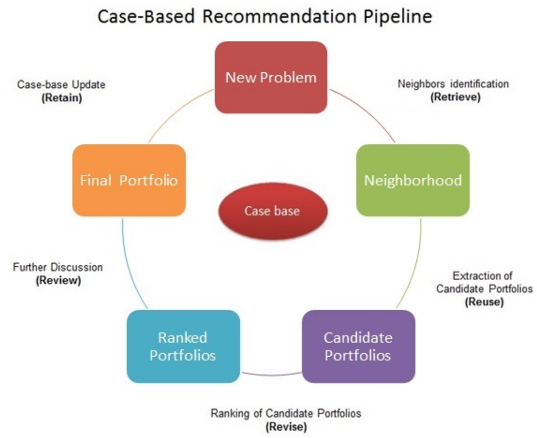

Figure 1. Case-Based Reasoning for Personalized Wealth Management

(1) Retrieve and Reuse: retrieval of similar portfolios is performed by representing each user through a feature vector: risk profile, inferred through the standard *MiFiD* questionnaire[5], investment goals, temporal goals, financial experience, and financial situation have been chosen as features. Each feature is represented on a five-point ordinal scale, from very low to very high. Next, *cosine similarity* is adopted to retrieve the most similar users (along with the portfolios they agreed) from the case base.

[1] Dipartimento di Informatica, Universita degli Studi di Bari "Aldo Moro", Bari, Italy, email:{cataldo.musto, giovanni.semeraro}@uniba.it

[2] The Risk Profile is defined as "an evaluation of an individual or organization's willingness to take risks". Typically, this value is obtained by conducting the above mentioned standard MiFiD questionnaire.

[3] http://en.wikipedia.org/wiki/Credit_rating

[4] http://en.m.wikipedia.org/wiki/Portfolio_(finance)

[5] http://en.wikipedia.org/wiki/Markets_in_Financial_Instruments_Directive

(2) Revise: *candidate solutions* retrieved at step 1 are typically too many to be consulted by a human advisor. Thus, the Revise step further filters this set to obtain the final solutions. To revise the candidate solutions, four techniques are compared:

(a) Basic Ranking: portfolios are ranked in descending *cosine similarity* order, according to the scores returned by the RETRIEVE step. The first *k* portfolios are returned to the advisor as *final solutions*.

(b) Greedy Diversification: this strategy implements the diversification algorithm described in [3]. The algorithm tries to diversify the *final solutions* by iteratively picking from the original set of *candidate solutions* the ones with the best compromise between *cosine similarity* and *intra-list diversity* with respect to the previously picked solutions. At each step of the strategy, the solution with the best compromise is removed from the set of candidate solutions and is stored in the set of final solutions.

(c) FCV: *Financial Confidence Value (FCV)* calculates how close to the optimal one is the distribution of the asset classes in a portofolio, according to the average historical yield obtained by each class. Given a set of asset classes A, for each portfolio p the set P, of the asset classes in it, and its complement $\overline{P}$ are computed. Next, FCV is formally defined as:

$$FCV(p) = Y(p)^{log(\lambda)+1} \tag{1}$$

$$Y(p) = \sum_{i=1}^{|P|} p_{a_i} * y_{a_i} \qquad \lambda = \frac{\sum_{i=1}^{|P|} y_{a_i}}{\sum_{k=1}^{|\overline{P}|} y_{a_k}} \tag{2}$$

where p_{a_i} and y_{a_i} are the percentage and the average yield of the *i-th* asset class in the portfolio, respectively. $Y(p)$ is the total yield obtained by the portfolio, and λ is a drift factor which calculates the ratio in terms of average yield between the asset classes in the portfolio and those which are not in. For values of $\lambda \geq 1$, it acts as a boosting factor (for $\lambda \ll 1$, it acts as a dumping factor). Through this strategy, all the *candidate solutions* are ranked according to the FCV score and the*top-k* solutions are returned to the advisor.

(d) FCV + Greedy: this combined strategy first uses the greedy algorithm to diversify the solutions, then exploits the FCV to rank the portfolios and obtain the *final solutions*.

(3) Review and Retain: in the Review step the user and the human advisor can further discuss and modify the portfolio, before generating the final solution for the user. If the monthly yield obtained by the newly recommended portfolio is acceptable, the solution is stored in the case base and can be used in the future as input to resolve similar cases.

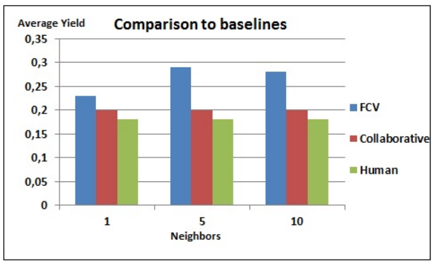

Figure 2. In vitro evaluation

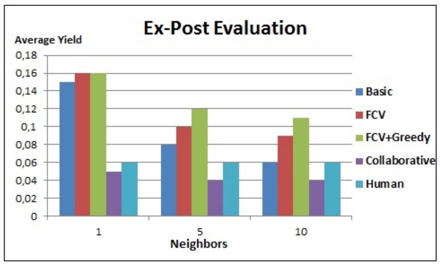

Figure 3. Ex-post evaluation

The performance of the framework has been evaluated in an experimental session against 1,172 real users. Results show that the yield obtained by recommended portfolios overcomes that of portfolios proposed by human advisors in many experimental settings. As shown in Figure 2, FCV significantly outperforms human recommendations (the average monthly yield increases from 0.18 to almost 0.30) for all the neighboorhood (put on the *X axis*) taken into account. The experimental results were further confirmed by an ex-post evaluation performed on real financial data from January to April 2014. As shown in Figure 3, this experiment provided very interesting results: beyond confirming the goodness of FCV-based ranking and the statistically significance of the gap with respect to both collaborative and human baselines, the most interesting outcome was that the combination of the diversification technique and FCV can further improve the performance of the proposed portfolios. This result suggests that the integration of the approaches can make the framework even more effective. This is due to the fact that a combined strategy can merge the advantages of a ranking based on past performance, as FCV, with an algorithm that may lead to more diverse recommendations. This makes the investment strategy better, since the human advisor does not base her investment proposal on a set of very similar portfolios, but rather on a set of *diversified solutions* which is more stable and effective, especially when market fluctuations have to be tackled.

4 Deployment of the framework

A demo version of the platform is available online[6].

Given that the platform is supposed to be of aid for financial advisors, it lets the advisor to select the current user as well as the recommendation technique to be adopted. Next, the "Recommendation" button shows the most promising portfolios for the target users along with the distribution of the asset classes. The distribution can be further discussed by user and advisor before coming to the final proposal which is stored in the case base.

REFERENCES

[1] P. Lops, M. de Gemmis, and G. Semeraro, 'Content-based recommender systems: State of the art and trends', in *Recommender Systems Handbook*, pp. 73–105. Springer, (2011).

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

[3] B. Smyth and P. McClave, 'Similarity vs. diversity', in *Case-Based Reasoning Research and Development*, 347–361, Springer, (2001).

[6] http://193.204.187.192:8080/OBWFinance/ - Login: 2 - Password: 12345

PsyRec: Psychological Concepts to enhance the Interaction with Recommender Systems

Gerhard Leitner[1]

Abstract. Although recommender systems are already a successful part of many online systems, there are still areas of research which are unexploited. One of them is the appropriate consideration of psychological theories which could be beneficial for the interaction between a computerized system and an online consumer, particularly in the financial services sector. This paper emphasizes the potentials of integrating psychological knowledge into the further development of recommender systems on the basis of psychological theories and basic decision processes. The enumerated concepts have been demonstrated to be influential in consumer buying behaviour in numerous studies and therefore are used as a theoretical basis of the presented work. A conceptual framework is build upon the technology acceptance model (TAM) which offers the possibility of integrating psychological knowledge in the further development of online financial services. Possible applications and implementations are shown on the basis of empirical work that has been carried out in the past years.

1 Introduction

The utility of recommender systems to enhance the quality of decision processes and their outcome has been approved many times, according to [1] they are among the most successful applications in Artificial Intelligence. Although recommenders have such a successful history, there are still unexploited potentials for advancement [2, 3]. Specifically promising in this regard is knowledge from psychology and research aiming to integrate it into recommender systems. This area of research is, taking the words of [4], still in its *infancy*. This paper opens new perspectives on the potentials of psychological concepts and theories to enhance the interaction with recommender systems in general and in the context of financial services in particular. The emphasis is put on interface and interaction aspects, because recommender systems are typically characterized by highly sophisticated algorithmic and technical basis. However, investigating also efforts in the enhancement of the interface is important, or, as Louis [5] formulated it: *"No matter how good your back-end systems are, the users will only remember your front end. Fail there and you will fail, period."*

The rest of the paper is structured as follows. In the first sections an introduction into the theoretical background with an emphasis on psychological concepts is given. This part is followed by a detailed discussion on decision phenomena and how these are related to recommender systems. Afterwards a framework based on the TAM, the *technology acceptance model* [6] is presented serving as a research basis for future research activities. In Section 6 studies which were carried out and showing concrete possibilities for combining psychological knowledge and recommender technologies are exemplified. The paper concludes with a discussion and an outlook on future work.

2 Theoretical Background

In the history of online sales many examples of online platforms exist which were characterized by high technical quality and innovativeness but lost market share or even disappeared because they did not appropriately consider user needs. For example, the first company offering books online was superseded by competitors who provided better user experience. Another example showing the importance of considering user needs is Boo.com, which was based on cutting edge technology but showed bad usability, see, for example, [5]. Recommender systems can be considered as state of the art technologies supporting online interaction and purchase and have demonstrated their benefits and capabilities in numerous studies. However, as [7] pointed out, decision support tools such as recommender systems consist of three parts: *"...database management capabilities, modelling functions, and a powerful yet simple user interface.."*. Specifically the latter offers high potentials for enhancement, by considering human capabilities such as attitudes, emotions, and other factors influencing their behaviour in their design. The goal to achieve is an enhanced quality of interaction between the human user and the computerized part of a system resulting in a better outcome for both, the user and the provider.

Recommender systems can be seen as the technical counterpart of real shopping environments. For about a century research in consumer psychology has been influential in advertising, marketing, and sales. Speaking of the *offline world* it does not surprise any more that the design of supermarkets in regard to shopping paths, lighting conditions or sound exposure is not left to chance and consumer psychology is omnipresent [8]. In comparison, psychological knowledge applied in the online sector is limited, although an increased consideration could be beneficial on different levels [9]. Specifically phenomena addressed in consumer and decision psychology are of interest in this regard [10, 11]. The challenge addressed in this paper is to take this knowledge to optimize recommender operated platforms in a way that consumers can, on the one hand, benefit from the advantages of information and communication technologies (ICT). This is possible because recommender systems are able to dynamically adapt to the individual user. This can constitute a meaningful alternative to offline purchase situations where an average sales assistant can be assumed to base his recommendations only on a limited set of alternatives. On the other hand it is important to make the user forget about the disadvantages online systems could have compared

[1] Alpen-Adria-Universität Klagenfurt, Institute for Informatics-Systems, Universitätsstrasse 65-67, 9020 Klagenfurt, Austria, email:gerhard.leitner@aau.at

to real shopping experiences. These are, for example, the possibility to touch and investigate a product physically and to communicate with a human counterpart, negotiate a price or ask questions. The challenge for the service-provider is the increased difficulty to convince an online user about the benefits of a product or even persuade him or her to buy it, because there are limited possibilities to establish a pleasant atmosphere. In the following a spotlight is put on a selection of psychological concepts and theories which have a direct relation to buying behaviour and therefore build a promising basis for further research and to enhance recommender systems in a way that they are capable of supporting all facets and phases of human consumer behaviour. This is neither easy nor possible in just one iteration.

3 Basic Psychological Theories

The following list of theories is not intended to be exhaustive, it should just point out the potentials of psychological concepts which have, as demonstrated in numerous studies, a direct relation to human behaviour and insofar could also be useful for the enhancement of online behaviour in general and in regard to financial services in particular. Some of the elements of the theories have been either analysed for applicability or actually used within own studies [12, 13, 1], others are planned to be integrated in our future work.

- **Prospect Theory, PT**
  PT is of interest in regard to the behaviour of consumers in situations characterized by uncertainty and and risk. These are, when considering the work of [10] demonstrating that the assumptions of economic theory do not hold, almost all situations. Because of limitations in human information processing, systematic biases in rating situations and decision making are observable. For example, humans act risk seeking when a loss is probable, or risk averse when a profit can be expected [11, 14]. This asymmetry is, for example, one explanation why people invest additional money into loss-making investments.
- **Locus of Control Theory, LoC**
  LoC implies that behaviour depends on the interpretation of a person whether she has control over a situation or interaction and the outcome of an interaction (internal locus of control). When a situation or outcome is beyond influence (e.g. the user has the feeling that the system or external forces have the control), then external locus of control is the case [15].
- **Attribution Theories, AT**
  Attribution theories are, as LoC, assuming internal/external control as one important dimension, but also include other dimensions, for example stability vs. flexibility. It is not only of relevance whether control is perceived as internal or external but also if it is stable, depending on the domain or a particular situation [16, 17]. An example for the influence of LoC and AT in the context of financial services is that a person may assume that it makes sense to actively control her financial portfolio (internal control) to increase prosperity. A person who observes herself as externally controlled may think that anyway only governments with taxation policies and financial service providers are responsible for the financial status of the individual. This attitude can be stable or flexible, the latter, for example, by observing the own financial situation as depending on the global economy and the possibility to change when the financial crisis is overcome.
- **Expectancy-Value Theories, EVT**
  This group of theories is based on the two dimensions expectancy and value. Expectancy refers to the degree to which a person is capable of reaching a goal. Value refers to the importance the goal has for the person. Example theories of this group are the theory of planned behaviour (TPB) or the theory of reasoned action (TRA) and they are important in the context of online buying. Besides personal aspects (i.e., attitude to a behaviour), social aspects play an important role and influence the value. For example, how people from relevant groups such as peer groups, family and friend would judge a certain behaviour (e.g., the purchase of a certain product) [18, 19].
- **Need for Cognition / Elaboration Likelihood Model, NfC**
  NfC implies that depending on the importance of the domain ("personal involvement") a person tends to process information on different elaboration *routes*. In domains which are of high importance for the person information is processed on the central route, characterized by a high level of elaboration (extensive collection of information, comparison, outweighing of pros and cons, etc.) The alternative way of processing, the peripheral route, is characterized by low involvement of the person and, as an effect, an intentional low investment of efforts in processing information. The type of elaboration is, for example, of interest when an online platform is intending to include persuasive technologies [20, 21].
- **Cognitive Dissonance, CD**
  CD is assuming a mental model that a person establishes about a certain area of life, a behaviour or other relevant issues. The model only includes "consonant" information, which means that information present in the model should not be contradictory. For example, if a person thinks about financing a holiday trip with a loan this may contradict with a negative attitude towards taking out a loan for things that do not have a material value (such as cars or real estates) . In this case dissonance occurs and, according to the model, mental efforts are invested to restore consistency [22]. For the concrete example an argument could be that the exchange rate of country's currency where the journey is heading is favourable and insofar money is saved.
- **Reactance Theory, RT**
  Implies that humans are driven by the assumption that they can behave and act unrestrictedly. If a behaviour or an *"object of desire"* is not available or difficult to reach, its subjective value is increased and the reactant user tries to overcome this shortage by increased efforts [23]. Online platforms try to induce reactance by indicating limitations in product or service availability. In regard to financial services, for example, special offers for loans or financing models are made available for limited time periods.
- **Flow, F**
  The central concept of the theory is the state of *flow* which is characterized by an immersion of the user with the system. Flow is, for example, observable on computer game players, musicians or craftsmen who smoothly interact with their tools without observable disruptions [24]. A platform offering financial services should aim at supporting flow by enabling a smooth interaction dialogue between user and system and giving the possibility to "play" with alternatives.

How elements of the enumerated theories and concepts could affect the interaction with a financial services platform is illustrated in the following example.

Example. *Imagine a potential consumer is using an online system to inform herself about loan opportunities. Based on her attributional patterns (AT, LoC) she has a certain understanding of whether she is able to use an online platform and can control the outcome of* *the product search. We assume that she is self-confident in the usage of the system (EVT, expectancy) and the system is appropriately designed that she can "play around" and easily evaluate alternatives (and eventually reaches a kind of "flow", F). Depending on the personal importance (EVT, value) of the product she is searching for (loan for a holiday trip, a car or a house) she will put low or high efforts in the evaluation, comparison, and selection of the product (NfC). When she knows what she wants and has good experiences with a certain brand or provider (PT, CD) she will not care that much what others say about her decision (EVT, peers). If she is uncertain, doesn't want to make a mistake or wants a product with a high status she will orient herself on information of other users (EVT, peers) and in what percentage they purchased what product (for example based on online ratings or discussions with her peer groups). If the product or service she has finally chosen is not available immediately, she will try to solve the problem by finding other sources from where to get the product (PT, RT) or she will resign and decide not to buy any product (AT).*

4 Decisions as the Connecting Element

The direct application of the theories and concepts enumerated above is difficult because many of them are too abstract. It is therefore necessary to investigate the "atomic" element of consumer behaviour which is *decision*. Each purchase or even browsing for information to prepare a purchase is characterized by a singular decision or a sequence of decisions. They are made on the basis of gathered information, the consultation of different information sources, the outweighing of alternatives, etc. Economic theory has assumed that humans can be considered as *omniscient* and make decisions on the basis of *optimal rationality*. Since the work of Simon [10] it is commonly agreed that this assumption does not hold for most decision situations. The majority of human decision processes is characterized by limited information use, biased mental models and routines either because of missing capabilities or a low level of motivation to invest cognitive efforts. Depending on the kind of limitation, technological means supporting the basic decision processes have to be designed in different ways.

Felser [25], based on the work of [26], categorizes decisions in consumer behaviour into 4 types, namely *extensive, limited, habitual and impulsive* decisions. What type of decision is actually applied is depending on the type of product or service, the degree of personal involvement, and emotional contribution (activation) to the domain and other personality traits. For example, searching for an appropriate loan for an apartment can have very different characteristics and motives.

Extensive Decision. If a person is planning to buy the apartment this is a long term investment that influences the financial life of the person for decades. Therefore the person is probably highly involved, activated, and will invest high efforts to find out the best financing alternative and therefore applies an *extensive* decision procedure until he gets the best financial plan which the smallest influence in the current financial situation. The strategy followed has characteristics of the central route processing of need for cognition theory [20, 21]. Although this type of decision making is highly sophisticated, it has some weaknesses. For example, the amount of information considered in the decision is not directly proportional to the amount of information available, which means that even if higher amounts of information would be available, people prefer short cuts [25]. An empirical proof for this hypothesis could be shown in our own work [1]. Another insight is that higher effort invested into a decision does not mean that the outcome of the decision is better. One of the reasons is that the dimensions consulted for a decision are often unconscious. An a posteriori justification is done on dimensions which can be rationalized but those may not be the ones which were responsible for the decision.

Limited Decision. Another person having in mind to rent an apartment and just needs money for new furniture may be less passionate and would apply other criteria to the decision process. She applies the second type of decision, which is limited decision. Decisions following this strategy are based on experiences (positive and negative ones) and heuristics which were derived from these experiences, such as *"Brand A is better than brand B"* or, *"The more expensive, the better a product"*. The person may choose the company for financing furniture based on an advertisement she recently saw. In this case the availability heuristic, described by [11, 14], is applied (e.g., brands and companies that are commonly known are better). Following this heuristic could lead to choosing a financing the furniture shop offers to his customers (an alternative the first person probably would not think about). An influence could also have the social environment (subjective norm, [18, 19]). Recommendations of relatives or friends which have good experiences with a bank can be taken into account.

Habitual Decision. The third type of decision, habitual decision, can be seen as a combination of extensive and limited decision. Based on previous experiences a mental model has been established, on the basis of which consumer behaviour follows a routine sequence and may not involve explicit decisions. This strategy mainly is applied in routine behaviour when no extraordinary investment is planned (such as in the previous examples). For example, if a person has to transfer money to a country where the receiver still requires conventional paper based transfer, she typically goes to her familiar bank branch and transfers the money there although there might be another company who offers cheaper transfers to the target country. In the past the selection of the best bank might have involved extensive decision strategies. When these efforts were successful and resulted in selecting an appropriate bank, a mental model is build which drives future behaviour. If the combination of services, price and reputation has been working satisfactorily in the past it would not have a serious impact, if it did not work any more (e.g., prices for services are slightly increased) - in terms of financial loss or well-being.

Impulsive Decisions. The last form - impulsive buying - is characterized as a "reaction" to environmental stimuli rather active behaviour and may not include decisions at all. This form of occurs in the context of financial services, for example, when a credit card is used for buying things. This also involves investing money, but the investment is hidden and partly unconscious.

The previous paragraph was describing decisions on a general level. Beckett et al.[27] have focused their work on financial products and present their findings in the form of a four-field decision matrix which has parallels to the four types of decisions described by [25]. Additionally to involvement, which is part of the systematic of [26, 25] and NfC [21], the authors point out confidence as another relevant dimension, which is a relevant dimension in LoC and AT [17] as well as the EVT [18]. The first decision type included in the matrix is *repeat-passive* decisions - which correspond to habitual decision in the nomenclature of [25]. Based on positive experiences the consumer has developed *loyalty* to an enterprise (a bank or insurance) and does not explicitly search for alternatives. The *rational-active* decision type corresponds to the extensive decision strategy. The third type identified by [27], *relational-dependent* decisions corresponds to [25, 26]'s limited decision type and is based on heuristics regarding experience and brand. If this strategy has been successful, *trust* is developed which reduces search and information processing activities. Finally, the impulsive type of [25] does not occur very often in the context of financial decisions. Therefore the matrix of [27] includes a fourth field labelled *"no purchase"*. Figure 1 is showing the decision types of [25] and their counterparts described in the work of [27].

Habitual Repeat-Passive	Extensive Rational-Active
Impulsive No Purchase	Limited Relational-Dependent

Figure 1. Comparison of decision types of [25] and [27]

The matrix has been evaluated in a series of focus groups and three product types are corresponding to the different decision types shown in Figure 1: *basic transaction services* (existing accounts), *basic insurances products* (car, house), and *investment services* (stocks, shares, pensions, etc.). Repeat-passive decisions mainly take place in the context of basic transaction services, when brand loyalty to banking institution and confidence in the decision is high. Rational-active decisions are made when price is one of the most important criteria. This strategy is characterized by the necessity to search for products, to deal with a big amount of information and to thoroughly analyse the outcome. This could be necessary because, for example, insurance companies offer more or less the same services and products and deliberately make comparison to competitive products difficult. Relational-dependent decisions are, according to the results achieved by [27] still strongly depending on personal communication and advice, because of the inherent complexity of the products and services.

The previous paragraphs were devoted to the *content* of decision processes involved in consumer behaviour. The second, similarly important dimension in regard to online platforms based on recommender systems is the *presentation* of information. We take the differentiation of [9] who proposes to differentiate two roles an online consumer has to assume, one as a *shopper* and the second as a *computer user*. What characterizes and drives the shopper has been emphasized above, in the next part the focus is put on the role of a computer user. Supporting a user in decision making requires the provision of interfaces that is appropriate, an issue the research areas of human computer interaction (HCI), usability engineering and user experience [28, 29, 30, 31] are dealing with. In regard to online consumer behaviour one of the major goals has to be to design interfaces in a way that they compensate the limitations an online system has in comparison to a to real world shopping situation and emphasize the advantages online systems have over real world shopping. The flexibility, adaptiveness, and adaptability of recommender systems enabling an individual support of each consumer is probably not available in typical shopping environments and insofar bear high potentials but are also challenging in regard to user interface design. This means, for example, that the development has to be based on state of the art interface design technologies, such as responsive design [32] and mobile first [33]. Not only the technology in the back-end (the recommender system) has to be adaptive, but also the interface itself should adapt to the needs of users. Burke [34] proposes a hybrid solution for recommender system technology, a similar approach could also be imagined for the user interface part. A one fits all approach seems not to be contemporary, different interface alternatives seem to be a proper way to provide an adaptive access to a recommender system for different groups of users in different contexts of use. One and the same user could be interacting with different views of the system, on different devices, depending on the task at hand, contextual aspects, and psychological factors such as involvement in the domain. This means that interfaces do not only have to be adaptive, but personalized, platform independent and customizable [35, 36]. The application of conventional usability engineering methods to accompany the development is crucial [37, 38], integrated in a user centred design process and combined with frequent evaluations involving representatives of the intended user groups.

5 An Integrated Model as Basis of Research

The aspects addressed in the previous sections characterizing consumer behaviour in general and online consumer behaviour in particular are difficult to capture. Their comprehension would be easier if a way could be found to operationalize them based on an integrated framework. The technology acceptance model (TAM) originally proposed by Davis [39] could build a basis for this attempt. TAM and its derivates have been empirically validated in numerous studies, and it optimally combines the two dimensions emphasized in the previous section. Content - meaning the psychological aspects related to a decision making and Presentation - aspects that related to human computer interaction. The TAM has relations to many of the theories and concepts enumerated in the previous sections. Figure 2 shows an adapted version of the latest version of TAM, TAM 3, introduced by [6]. The dimensions of TAM and their relation to the concepts and theories enumerated above are described in this section. The descriptions are partly taken from [6, 40].

- **Experience**
  Already having used a system or similar ones can have an influence on many factors, such as the perceived usefulness and the subjective norm. In relation to psychological theories, experience can increase, for example, the confidence and the assumption of internal control (LoC, AT).
- **Voluntariness**
  The extent to which users perceive the usage of a system to be non-mandatory. This aspect relates to reactance theory (RT) - if a person has the freedom to choose an online system for financial services additionally to offline services this makes a difference to being forced to use online services (because the nearby bank branch has been closed).
- **Subjective Norm**
  A person's perception that most people who are important think he or she should or should not perform a behaviour or use a system. There could, for example, be a conflict between the personal preferences and the attitude of the relevant others, which could lead to cognitive dissonance (CD) ("I would issue a credit for a holiday trip".)
- **Image**
  The degree to which the use of an innovation is perceived to enhance one's status in the social system. In regard to the provision of different platforms (desktop or mobile platforms) this aspect,

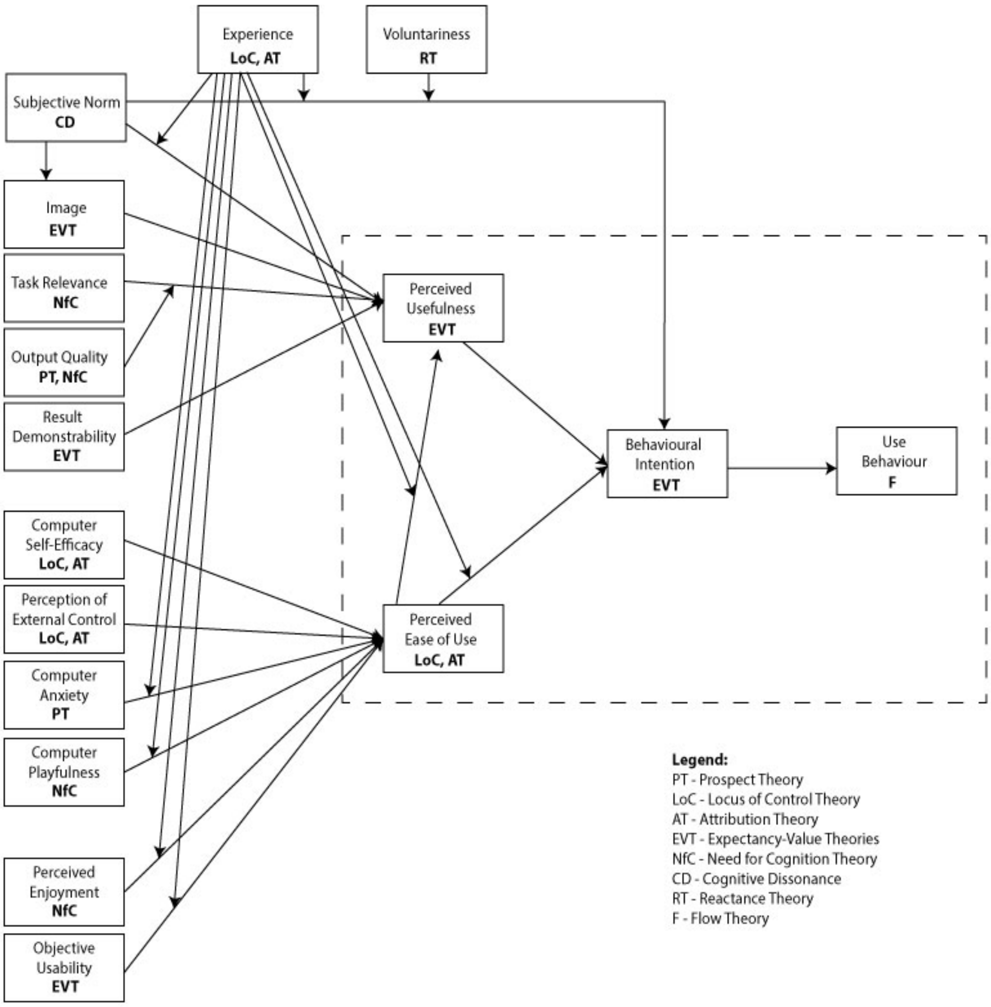

Figure 2. Technology Acceptance Model Version 3, adapted from [40] and complemented with example relations to psychological theories

for example, influences the usage of a mobile app. It is depending on whether or not the platform is accepted by the peer group (Apple, Android, Windows mobile) and illustrates that the attitude towards a system is not always based on functional requirements (EVT).

- **Task Relevance**
  A person's perception regarding the degree to which the target system is relevant to his or her life. If a system offers enhanced efficiency (e.g., not having to visit a bank branch for basic tasks) without loosing quality (NfC) it will be used.
- **Output Quality**
  The degree to which a person believes that the system offers the same services and enables to achieve the same results as other alternatives, for example, services offered in a bank branch (PT, NfC).
- **Result Demonstrability**
  Tangibility of the results of using the system. This aspect has relations to subjective norm and image, for example showing increased prosperity as a result of intelligent investments (EVT).
- **Computer Self-Efficacy**
  The degree to which a person beliefs that he or she has the ability to perform the intended task. This depends on the experience with computer systems in general, and on the experiences within a specific domain (e.g. financial services) in particular (LoC, AT).
- **Perceptions of External Control**
  The degree to which a person believes that an organizational and technical infrastructure exists to support use of the system. This could also be influential in a negative way (according to LoC and AT) when a person feels that the organization behind a system limits his or her performance or degrees of freedom.
- **Computer Anxiety**
  The degree of a person's fear, when she/he is faced with the need of using computers to access services. Specifically in the context of financial services (or even online transactions with credit cards) people are anxious because of the danger to lose money (PT).
- **Computer Playfulness**
  The degree of cognitive spontaneity in computer interactions. If a system supports this kind of interaction, such as simulating differ-

ent variants of financing, this supports persons engaging in extensive decision making processes (NfC).

- **Perceived Enjoyment**
The extent to which using a specific system is perceived to be enjoyable, whereas enjoyment can have different dimensions. Feeling safe in the sense of nothing unexpected can happen when transferring money could be one form of enjoyment. Another one is developing trust towards an institution or a platform when the latter is characterized by transparency and comprehensibility (NfC).
- **Objective Usability**
A comparison of systems based on the actual level of effort required to complete specific tasks. If it is faster to go to the bank branch to transfer money than using the computer interface, then the objective usability of an online system would be low (EVT).
- **Perceived Usefulness**
The degree to which a person believes that using the system will help him or her to attain gains in life quality. Saving money by using an online system instead of personal services convinces people to adapt to new technologies (EVT).
- **Perceived Ease of Use**
The degree of ease associated with the use of the system. Besides the utility aspects of a system, the subjective usability is relevant. If people do not trust a system or are doubtful in their usage, they would not use it (LoC, AT).
- **Behavioural Intention**
The degree to which a person has conscious plans to perform or not perform some specified behaviour. Only if the enumerated dimensions are fulfilled in a certain degree, a person will have the intention to use a system. The correlation between the intention and the actual use still is low (EVT).
- **Use Behaviour** When every aspect is, depending on the individual preferences, optimally fulfilled, then a flow experience could occur (F).

As emphasized in the enumeration of elements, the TAM has connections to the concepts and theories addressed in this paper [9] and would also allow the integration of additional aspects, for example trust, cf. e.g. [41, 42, 43, 44]. The TAM has also served as basis for research in the financial services domain, cf. e.g. [45, 46, 47].

6 Empirical Work

The theoretical concepts presented in this paper have been evaluated in several empirical works. In this section a selection of these works and their relation to the theoretical parts of the paper is presented and relations to the enumerated models and concepts are emphasized.

The first work in this regard is a paper on serial position effects. The effect, being one of the oldest phenomena in psychological basic research [48, 49, 50], is characterized by the fact that items presented in a list or sequence are better memorized when presented at the beginning or the end of the list. In our work [1] we could show that changing the sequence of items significantly influences the recall of the items and this offers a possibility to influence the interaction between a consumer and a computer system on the level of presentation. Depending on the motives and needs that drive the consumer (e.g. involvement, confidence, type of decision, willingness to invest efforts) important information can be put in the sequence where it has the highest probability to be perceived and memorized for further usage. Figure 3 is shows the effect on the recall of items by simply changing their order. The list used in the study contained features of digital cameras (pixels, storage, zoom). Only the order of items was manipulated but this significantly increased their recall.

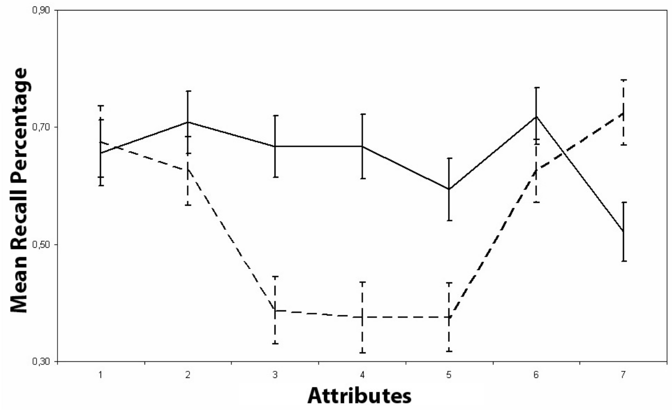

Figure 3. Recall frequency in a manipulated item sequence (continuous line) and a familiar item sequence (dashed line) [1]

A more recent work which builds upon the work on serial position effects was carried out in the domain of group decision making [52]. Making decisions in groups, for example choosing a dinner with a business partner or deciding what movie to watch with friends in a cinema always involves psychological phenomena on the individual as well as on the group level. Decisions derived in group situations are influenced by rhetoric skills of the participants, negotiation techniques applied, leadership competency and other personality factors. In contrast to this real-time and synchronous approach, an online tool supports asynchronous and sequential decision procedures. Psychological concepts that could have an impact in this kind of decision process are, for example, originating from research groups who developed the prospect theory [11, 14]. One group of effects are *anchoring* or *framing* effects, or more general, *context* effects [53, 51]. A following small example illustrates their influence. To be able to sketch a financial plan it is necessary to have a starting point, the anchor stimulus. This starting point is typically the amount of money that has to be financed. A strategy that is frequently used in advertising is not to use the whole amount for evaluation (for example, 100.000 are needed + overhead costs) but the monthly rate (for example 500). Within the study we investigated alternatives of presenting information and were interest in the possibilities of manipulating serial position effects and other form of presentation, concretely based on the multi attribute utility model (MAUT). The results showed that MAUT concepts can counteract serial position effects and insofar represent an appropriate means to steer decision processes. Figure 4 is showing an example screen of the CHOICLA group decision support tool on which preferences can be declared based on multiple attributes.

The last empirical work presented was focused on persuasion [54] and the potentials of the asymmetric dominance effect, better known as *decoy* effect [55]. This concept has also a relation to anchoring and framing effects which can be manipulated. In contrast to the example above where information is hidden or presented in another form, the decoy effect uses the influence of adding additional information to a decision situation. Adding a decoy element is intended to divert or even disturb the attentive processes of a potential consumer and open a new perspective to him or her to lead a decision in a certain

Figure 4. Choicla Screen to enter preferences for restaurants based on MAUT [52]

direction, to persuade a user to purchase a product or to initiate a preference construction which would not have been started without the distractive element. In our paper we investigated the asymmetric dominance effect and could show possibilities how to integrate them into recommender systems. Figure 5 is showing a decoy situation. Before introducing the decoy element (D) two products are available to the customer, C (competitor product) and T (target product). C is characterized by a lower price, but also by lower quality than T. As price is one of the most important dimensions in purchase decisions [26] consumers tend to buy C. With introducing the decoy D which has a lower quality than T, but a higher price, the focus of attention is directed to quality. This new perspective is not only of advantage for the provider (because of higher revenue) but also for the consumer (because of higher quality and satisfaction with the product).

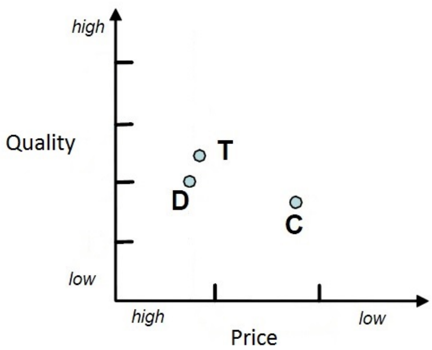

Figure 5. Showing the example for the asymmetric dominance (”decoy”) effect. Product C (competitor) is of lower quality than product T (the target product), but C is cheaper and price is typically the feature with the highest influence in purchase situations. People would therefore, in general, choose product C. By introducing a product D (decoy) which is of higher quality than C, but of lower quality than T and more expensive than both of them, the viewpoint (anchor, reference frame) changes, and product T is preferred by the majority of consumers [54]

7 Discussion and Conclusions

In this paper we have tried to emphasise the potentials of psychological theories to enhance the quality of interaction between users and computerised systems based on recommender technology. The theoretical basis builds a selection of psychological concepts and theories which have been empirically investigated in numerous studies and proved themselves as being relevant in the context of consumer behaviour. An increased consideration of knowledge from psychology could enhance the quality of recommender systems, specifically on the level of the user interface. The different types of decisions related to consumer behaviour were discussed and possibilities of recommender systems to support such decisions were exemplified. The technology acceptance model serves as a basis for further research in this area because it already integrates many of the relevant psychological concepts and theories that have been demonstrated to be influential in the context of consumer behaviour. With an appropriate consideration of this knowledge, recommender systems could overcome the disadvantages online system have in comparison to offline interaction between consumers and, for example, shop assistants. The advantages of recommender systems such as their capabilities of processing huge amounts of data, selecting the correct products from millions of alternatives, and calculating the best product for are consumer within a few seconds could be exploited in a better way if not only the back-end functionalities but also the front-end, the interface to the customer is enhanced in an appropriate way.

Although our work is addressing different domains, the conceptual work sketched and the empirical studies performed are also applicable to the financial sector. Specifically of interest in this regard are the different types of decisions driving potential customers and motivating them to use an online system, choosing a product or service, changing parts of his or her financial portfolio. In the context of recent developments in the financial sector (e.g., merging of banks and insurance companies, closing of branches) the importance of online services will increase. Appropriate systems supporting the different needs, motives of end consumers, and also respecting the different levels of efforts people are willing to invest into financial decisions will be more important than ever before. Recommender systems integrating psychological aspect and simulating a *”human image”* [36] could fill the arising gaps. With the system MYLIFE, an award winning platform, we could demonstrate respective possibilities. MYLIFE is an online platform enabling insurance agents together with end consumers to manage the consumer’s financial portfolio in a cooperative partnership instead of putting the consumer in the role of a *”supplicant”* towards financial service providers. The system consists of an intelligent algorithmic basis FASTDIAG [56] and an appropriate user interface visualizing in an integrated fashion the finance portfolio of a customer.

The empirical work presented can only be seen as the starting point in the endeavour of enhancing human recommender interaction in the emphasized way. An unresolved problem in this regard is, for example, how a recommender system could find out what strategy a consumer is currently applying (e.g. extensive or limited decision) and to change the presentation of information accordingly. There are of course domains where one strategy is the most probable one (e.g. financing a real estate are probably based on extensive and central route elaboration) but further research is necessary to address this problem. Of course transferring services form offline to online does not only have advantages. In the context of current developments in regard to privacy and business ethics this opens new challenges which are influencing the orientation of future research activities. Our major goal is to complete the ”puzzle” of which we have already identified elements in our past research work.

REFERENCES

[1] Felfernig, A., Friedrich, G., Gula, B., Hitz, M., Kruggel, T., Leitner, G., Melcher, R., Riepan, D., Strauss, S., Teppan, E., & Vitouch, O. (2007). Persuasive recommendation: serial position effects in knowledge-based recommender systems. In: Persuasive Technology, 283-294.

[2] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6), 734-749.

[3] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. Communications of the ACM, 40(3), 77-87.

[4] Chen, L., de Gemmis, M., Felfernig, A., Lops, P., Ricci, F., & Semeraro, G. (2013). Human decision making and recommender systems. ACM Transactions on Interactive Intelligent Systems (TiiS), 3(3), 17.

[5] Louis, T. Boo.com goes bust. The Boo.com post-mortem, from an insider. `http://www.tnl.net/blog/2000/05/19/boocom-goes-bust/`

[6] Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. Decision sciences, 39(2), 273-315.

[7] Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., & Carlsson, C. (2002). Past, present, and future of decision support technology. Decision support systems, 33(2), 111-126.

[8] Bell, S.J.(1999) Image and consumer attraction to intra-urban retail areas: An environmental psychology approach, Journal of Retailing and Consumer Services, Volume 6, Issue 2, 67-78.

[9] Koufaris, M. (2002). Applying the technology acceptance model and flow theory to online consumer behavior. Information systems research, 13(2), 205-223.

[10] Simon, H. A. (1955). A behavioral model of rational choice, Quarterly Journal of Economics 69, 99-118.

[11] Tversky, A., Kahneman, D. (1986) Rational Choice and the Framing of Decisions. Journal of Business, 59/4,Part 2, 251-278.

[12] Leitner, G. (1998) Stressfaktoren, Stresserleben und subjektive Attributionsmuster. Diploma Thesis, University of Vienna.

[13] Leitner, G., Mitrea, O., & Fercher, A. J. (2013). Towards an Acceptance Model for AAL. In Human Factors in Computing and Informatics 672-679.

[14] Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. Econometrica: Journal of the Econometric Society, 263-291.

[15] Rotter, J. B. (1990). Internal versus external control of reinforcement: A case history of a variable. American psychologist, 45(4), 489.

[16] Kelley, H. H. (1973). The processes of causal attribution. American psychologist, 28(2), 107.

[17] Weiner, B. (2012). An attribution theory of motivation. Handbook of theories of social psychology, 1, 135-155.

[18] Ajzen, I. (1991). The theory of planned behavior. Organizational behavior and human decision processes, 50(2), 179-211.

[19] Fishbein, M. (1979). A theory of reasoned action: some applications and implications. In: Howe et al.. 65-116.

[20] Smith, S., Levin, I. (1998) Need for Cognition and Choice Framing Effects, Journal of Behavioral Decision Making, 9(4): 283-290.

[21] Haugtvedt, C. P., Petty, R. E., & Cacioppo, J. T. (1992). Need for cognition and advertising: Understanding the role of personality variables in consumer behavior. Journal of Consumer Psychology, 1(3), 239-260.

[22] Festinger, L. (1962). A theory of cognitive dissonance (Vol. 2). Stanford university press.

[23] Brehm, J. W. (1966). A theory of psychological reactance. Academic Press.

[24] Csikszentmihalyi, M. (1990). Flow: The Psychology of Optimal Performance. Cambridge University Press.

[25] Felser, G. (2007). Werbe-und Konsumentenpsychologie. Spektrum.

[26] Weinberg, P., Gröppel-Klein, A., & Kroeber-Riel, W. (2003). Konsumentenverhalten. Vahlen.

[27] Beckett, A., Hewer, P., & Howcroft, B. (2000). An exposition of consumer behaviour in the financial services industry. International Journal of Bank Marketing, 18(1), 15-26.

[28] Donahue, G.M., Weinschenk, S., Nowicki, J. (1999). Usability Is Good Business. `http://half-tide.net/UsabilityCost-BenefitPaper.pdf`.

[29] Van Pelt, A., & Hey, J. (2011). Using TRIZ and human-centered design for consumer product development. Procedia Engineering, 9, 688-693.

[30] Lohse, G. J. L. (2000). Usability and profits in the digital economy. In People and Computers XIV Usability or Else!, 3-15.

[31] Lohse, G. L., & Spiller, P. (1999). Internet retail store design: How the user interface influences traffic and sales. Journal of Computer Mediated Communication, 5(2), 0-0.

[32] Marcotte, E. (2011). Responsive web design. Editions Eyrolles.

[33] Wroblewski, L. (2012). Mobile first. Editions Eyrolles.

[34] Burke, R. 2002. Hybrid recommender systems: Survey and Experiments. User Model. User-Adapt. Interact. 12, 4, 180-200.

[35] Karat, C. M., & Blom, J. O. (Eds.). (2004). Designing personalized user experiences in eCommerce (Vol. 5). Springer Science & Business Media.

[36] Blom, J. (2002). A theory of personalized recommendations. In CHI'02 Extended Abstracts on Human Factors in Computing Systems, 540-541.

[37] Weibelzahl, S. (2001). Evaluation of adaptive systems, In: User Modeling 2001. LNAI 2109, 292-294.

[38] Swearingen, K., & Sinha, R. (2001, September). Beyond algorithms: An HCI perspective on recommender systems. In ACM SIGIR 2001 Workshop on Recommender Systems, Vol. 13, No. 5-6, 1-11.

[39] Davis Jr, F. D. (1986). A technology acceptance model for empirically testing new end-user information systems: Theory and results. Doctoral dissertation, Massachusetts Institute of Technology.

[40] Venkatesh, V. Theoretical models of Acceptance `http://www.vvenkatesh.com/it/organizations/theoretical-models.asp`

[41] Gefen, D., Karahanna, E., & Straub, D. W. (2003). Trust and TAM in online shopping: An integrated model. MIS quarterly, 27(1), 51-90.

[42] Benbasat, I., & Wang, W. (2005). Trust in and adoption of online recommendation agents. Journal of the Association for Information Systems, 6(3), 4.

[43] Pavlou, P. A. (2003). Consumer acceptance of electronic commerce: Integrating trust and risk with the technology acceptance model. International journal of electronic commerce, 7(3), 101-134.

[44] Suh, B., & Han, I. (2003). Effect of trust on customer acceptance of Internet banking. Electronic Commerce research and applications, 1(3), 247-263.

[45] McKechnie, S., Winklhofer, H., & Ennew, C. (2006). Applying the technology acceptance model to the online retailing of financial services. International Journal of Retail & Distribution Management, 34(4/5), 388-410.

[46] Pikkarainen, T., Pikkarainen, K., Karjaluoto, H., & Pahnila, S. (2004). Consumer acceptance of online banking: an extension of the technology acceptance model. Internet research, 14(3), 224-235.

[47] Featherman, M. S., & Pavlou, P. A. (2003). Predicting e-services adoption: a perceived risk facets perspective. International journal of human-computer studies, 59(4), 451-474.

[48] Kirkpatrick, E. A. (1894). An experimental study of memory. Psychological Review, 1(6), 602.

[49] Ebbinghaus, H. (2013). Memory: A contribution to experimental psychology. Annals of neurosciences, 20(4), 155.

[50] Gershberg, F., Shimamura, A. (1994) Serial position effects in implicit and explicit tests of memory, Journal of Experimental Psychology: Learning, Memory, and Cognition, 20, 1370-1378.

[51] Simonson, I. Tversky, A. (1992) Choice in Context: Tradeoff Contrast and Extremeness Aversion, Journal of Marketing Research, 29. 281-295.

[52] Stettinger, M., Felfernig, A., Leitner, G., Reiterer, S., & Jeran, M. (2015). Counteracting Serial Position Effects in the CHOICLA Group Decision Support Environment. In Proceedings of the 20th International Conference on Intelligent User Interfaces, 148-157.

[53] Hamilton, R. (2003) Why Do People Suggest What They Do Not Want? Using Context Effects to Influence Others Choices, Journal of Consumer Research, 29, 492-506.

[54] Felfernig, A., Gula, B., Leitner, G., Maier, M., Melcher, R., & Teppan, E. (2008). Persuasion in knowledge-based recommendation. In Persuasive Technology, 71-82.

[55] Huber, J., Payne, W., Puto, C. (1982) Adding Asymmetrically Dominated Alternatives: Violations of Regularity and the Similarity Hypothesis, Journal of Consumer Research, 9, 90-98.

[56] Felfernig, A., Schubert, M. & Zehentner, C. (2012) An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets, Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM), Cambridge University Press, 26(1), 53-62.