iCone: intelligent environment for the Development and Maintenance of Configuration Knowledge bases

Franz Wotawa, Florian Reinfrank, Gerald Ninaus, Alexander Felfernig[1]
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b/II, 8010 Graz, Austria
{firstname.lastname}@ist.tugraz.at

Abstract

Constraint-based recommendation systems are used in many different domains like notebooks, cars, and mobile phones. Such systems describe product domains in sets of product and customer variables, their domains, and constraints which define the relationship between the variables.

Maintenance is a crucial task in constraint-based recommendation systems, because it is time-consuming and error-prone. We implemented a new application called 'iCone' (intelligent environment for the development and maintenance of configuration knowledge bases) to support knowledge engineers and their maintenance tasks. We present intelligent techniques like recommendation, anomaly management, dependency detection, and metrics to support knowledge engineers when maintaining constraint-based systems.

1 Introduction

Many different product domains like, computers, cars, or mobile phones use constraint-based recommendation systems as a model for modern e-commerce services. The model can be represented as a constraint satisfaction problem (CSP) and consists of products, questions for the customer, and constraints which are describing the relationships between products and questions.

In this paper we give an overview of the iCone interface which is a web-tool for the development and maintenance of constraint-based systems.[2]. This system can be used to create and maintain constraint-based systems like knowledge-based recommendation or knowledge-based configuration. The advantage of this system is a strong intelligent support of knowledge engineers when maintaining a constraint-based system which will be described in this paper.

2 iCone

iCone is a java-based web-application with a SQLite data base. SQL will be used to do consistency checks and to save and load the knowledge bases. The application has four main packages called '*KnowledgeBase*' (with the main class '*KnowledgeBase*'), '*Algorithm*' (calculating anomalies), '*Anomaly*' (e.g., explaining anomalies), and '*Solver*' (preparing consistency checks). For the visualization, the users need to have a modern internet browser with active JavaScript. All screens consists of *login*, *navigation*, *recommendation and notification*, and *main* area. In the main area, the user sees lists of knowledge bases, products, questions, constraints, results of anomaly checks or a preview. Figure 1 shows a preview of the constraint-based recommendation system for mobile phones. In the left area the user can answer some questions. Based on constraints which are defining the relationship between questions and products, the system recommends a set of products for the user.

The main object of iCone is the *Knowledge base*. It contains all products, product variables, questions, and constraints. Furthermore, iCone deals with the *analysis* package, which detects anomalies in the knowledge base, generates recommendations for knowledge engineers, approximates dependencies between variables in the knowledge base and generates metrics to evaluate the knowledge base. Next, we give an overview about the four main supporting techniques.

Recommendation

We implemented four different recommendation techniques.[3]

- user-independent recommendation techniques like **most viewed recommendation** and **recently added items**.
- user-dependent recommendation techniques like **collaborative filtering** (find peer users) and **content-based filtering** (find similar items)

Anomaly management

In our implementation we consider three different types of anomalies:[4]

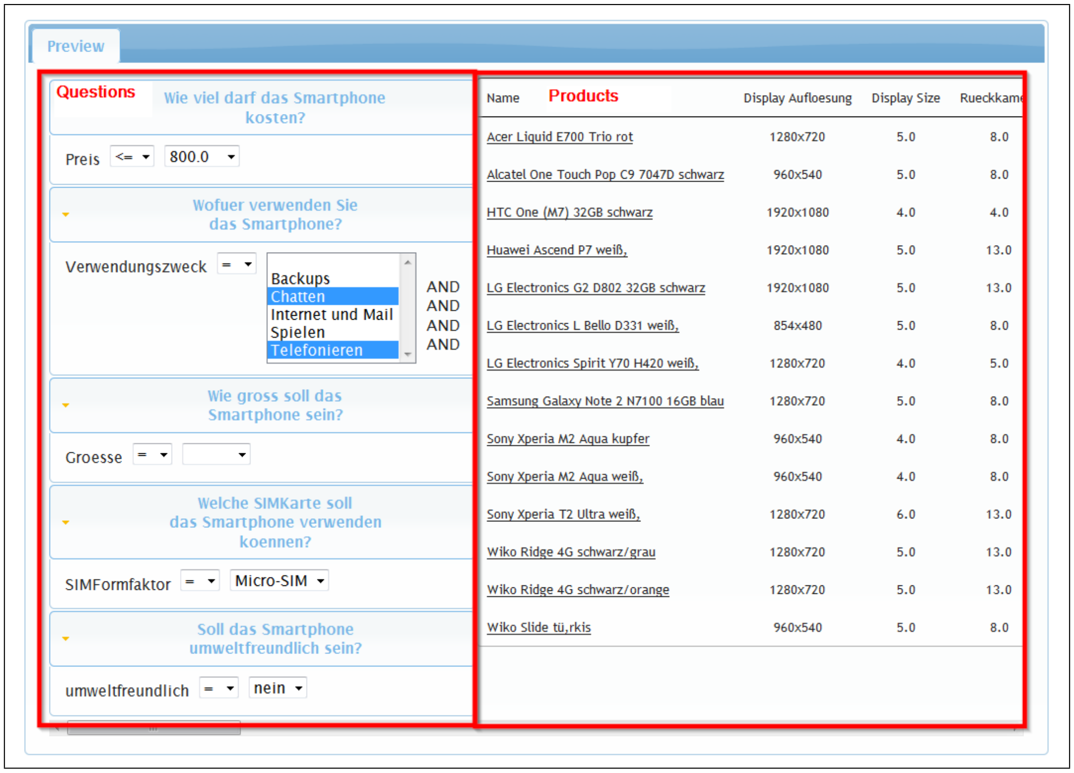
- **conflict**: A conflict is a set of constraints which can not be fulfilled. Conflicts can be resolved by sets of **diagnoses**.
- **redundancy**: A set of constraints can be denoted as redundant, if the removal of this set does not change the behavior of the knowledge base.

[1] The authors are ordered in reverse alphabetical order.

[2] `http://ase-projects-studies.ist.tugraz.at:8080/iCone/index.jsp`

[3] For a detailed description we refer the reader to [1; 2].

[4] For a detailed description we refer the reader to [2].



Wie viel darf das Smartphone kosten?

Preis <= 800.0

Wofuer verwenden Sie das Smartphone?

Verwendungszweck =
Backups
Chatten AND
Internet und Mail AND
Spielen AND
Telefonieren AND

Wie gross soll das Smartphone sein?

Groesse =

Welche SIMKarte soll das Smartphone verwenden koennen?

SIMFormfaktor = Micro-SIM

Soll das Smartphone umweltfreundlich sein?

umweltfreundlich = nein

Name	Display Aufloesung	Display Size	Rueckkame
Acer Liquid E700 Trio rot	1280x720	5.0	8.0
Alcatel One Touch Pop C9 7047D schwarz	960x540	5.0	8.0
HTC One (M7) 32GB schwarz	1920x1080	4.0	4.0
Huawei Ascend P7 weiß,	1920x1080	5.0	13.0
LG Electronics G2 D802 32GB schwarz	1920x1080	5.0	13.0
LG Electronics L Bello D331 weiß,	854x480	5.0	8.0
LG Electronics Spirit Y70 H420 weiß,	1280x720	4.0	5.0
Samsung Galaxy Note 2 N7100 16GB blau	1280x720	5.0	8.0
Sony Xperia M2 Aqua kupfer	960x540	4.0	8.0
Sony Xperia M2 Aqua weiß,	960x540	4.0	8.0
Sony Xperia T2 Ultra weiß,	1280x720	6.0	13.0
Wiko Ridge 4G schwarz/grau	1280x720	5.0	13.0
Wiko Ridge 4G schwarz/orange	1280x720	5.0	13.0
Wiko Slide tü,rkis	960x540	5.0	8.0

Figure 1: Preview of a constraint-based recommendation system for mobile phones

- **well-formedness violation**: Well-formedness violations don't change the behavior of the knowledge base but make it difficult to maintain a knowledge base.

Dependency detection

In our implementation we have visualizations for constraint dependencies and another one for variable dependencies.[5]

- **Dependencies between constraints** shows the relationship between products, question, and constraints based on content-based recommendation.
- **Dependencies between variables** will be either calculated (if possible) or approximated (in big knowledge bases). For the approximation we use Gibb's sampling.

Metric calculation and evaluation

To get an overview of the quality of the knowledge base, the iCone interface offers an overview of several metrics.[6]

- **Goal question metrics** (GQM): We indicated three goals, five questions and 16 metrics.
- **Function point analysis** (FPA): We adapted the five input variables (from the software engineering to the knowledge engineering domain) and present the results for each knowledge base in our application.

In this paper we presented recommendation techniques, anomaly management, dependency detection, and metric calculation, which are novel techniques for the maintenance of constraint-based systems.

3 Summary

In this paper we gave an overview of our iCone system. With iCone you can create constraint-based configuration, knowledge-based recommendation, and MAUT-based knowledge bases. You can create, read, update, and delete products, product variables, questions, and constraints.

Further enhancements are possible in the context of supporting collaboration (e.g. when two or more knowledge engineers are working on the same knowledge base) and in the support of test case generation.

References

[1] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*, volume 1. University Press, Cambridge, 2010.

[2] Florian Reinfrank, Gerald Ninaus, and Alexander Felfernig. Intelligent techniques for the maintenance of constraint-based systems. *Configuration Workshop*, 2015.

[3] Florian Reinfrank, Gerald Ninaus, Bernhard Peischl, and Franz Wotawa. A goal-question-metrics model for configuration knowledge bases. *Configuration Workshop*, 2015.

[4] Florian Reinfrank, Gerald Ninaus, Franz Wotawa, and Alexander Felfernig. Maintaining constraint-based configuration systems: Challenges ahead. *Configuration Workshop*, 2015.

[5] For a detailed description we refer the reader to [4].

[6] For a detailed description we refer the reader to [3].