Intelligent Techniques for Software Product Line Engineering

2nd International Workshop on Formal Methods and Analysis in Software Product Line Engineering

Univ.-Prof. DI Dr. Alexander Felfernig

Institute for Software Technology
Graz University of Technology, Austria
alexander.felfernig@ist.tugraz.at
Research @ ASE @ IST @ TU Graz

Recommendation Technologies for Complex Products & Services

Recommendation & Decision Technologies for Requirements Engineering

Intelligent Assistance for Managing Complex Constraint Sets

Intelligent Systems

Software Engineering

Decision Technologies

FLEXTRONICS
Persuasive Technologies for Software Development

wüstenrot

COHAVE
Consumer Behavior & Decision Modeling for Recommender Systems

PLAIN-IT

 ELSEVIER
Book: „Configuration Systems in Practice“

wüstenrot

ilogs

Recommender Systems in AAL

Research @ ASE @ IST @ TU Graz

Univ.-Prof. DI Dr. Alexander Felfernig

Research
Intelligent Systems

- **Applied AI**: Systems solving problems by simulating intelligent behavior. [Russel and Norvig, 2003]

- **Specific Type**: Constraint-based Systems. [Freuder 1997]

- **Examples**:
  - Planning
    - Production
    - Resources
    - Software Releases
  - Recommendation
    - Financial Services
    - Consumer Electronics
    - e-Tourism
  - Configuration
    - Automotive
    - Financial Services
    - Software Product Lines

- **Application**: CRM, Online Selling, Call Center, Corporate Memory, Company Web, Project Management
Example: Car Configuration

**Variables:** \( V = \{v_1, v_2, \ldots, v_n\} \)

**Domains:** \( D = \{d_1, d_2, \ldots, d_n\}, d_j = \text{dom}(v_j) \)

**Product Knowledge:**
\( C_{KB} = \{c_1, c_2, \ldots, c_q\} \)

**Customer Requirements:**
\( C_R = \{c_{q+1}, c_{q+2}, \ldots, c_s\} \)

**Configuration Task**

**Consistency Check**
- Completion
- Explanation

**Constraint Solver (Configurator)**

**Solution (Consistent Configuration)**
\( S = \{v_1 = a_1, v_2 = a_2, \ldots, v_n = a_n\}, a_j \in \text{dom}(v_j) \)

**Example:** Car Configuration

**Variables:** \( V = \{\text{type, pdc, fuel, skibag, 4-wheel, color}\} \)

**Domains:**
- \( \text{dom}(\text{type}) = \{\text{city, limo, combi, xdrive}\} \)
- \( \text{dom}(\text{pdc}) = \{\text{yes, no}\} \)
- \( \text{dom}(\text{fuel}) = \{1.7, 2.6, 4.2\} \)
- \( \text{dom}(\text{skibag}) = \{\text{yes, no}\} \)
- \( \text{dom}(\text{4-wheel}) = \{\text{yes, no}\} \)
- \( \text{dom}(\text{color}) = \{\text{red, blue, gray, black}\} \)

**Product Knowledge (CKB):**
- \( c_1: \text{4-wheel = yes } \rightarrow \text{type = xdrive} \)
- \( c_2: \text{skibag = yes } \rightarrow \text{type } \neq \text{city} \)
- \( c_3: \text{fuel = 1.7 } \rightarrow \text{type = city} \)
- \( c_4: \text{fuel = 2.6 } \rightarrow \text{type } \neq \text{xdrive} \)
- \( c_5: \text{type = combi } \rightarrow \text{skibag = yes} \)
- \( c_6: \text{type = limo } \rightarrow \text{pdc = yes} \)

**Customer Requirements (CR):**
- \( c_7: \text{type = city} \)
- \( c_8: \text{fuel = 1.7} \)
- \( c_9: \text{4-wheel = no} \)
- \( c_{10}: \text{pdc = yes} \)
- \( c_{11}: \text{color = black} \)

**Solution:**
\( S = \{\text{type = city, fuel = 1.7, 4-wheel = no, pdc = yes, color = black}\} \)

**Motivation**
Example: Feature Set Configuration

\[ \text{V} = \{ \text{Phone}, \text{Calls}, \text{GPS}, \text{Screen}, \text{Media}, \text{Basic}, \text{Colour}, \text{HighRes}, \text{Camera}, \text{MP3} \} \]

\[ \text{D} = \{ \text{dom(Phone)}=\text{dom(Calls)}= \ldots = \text{dom(MP3)}=\{\text{yes},\text{no}\} \} \]

\[ \text{C}_{\text{KB}} = \{ c_1: \text{Phone} = \text{yes}, c_2: \text{Phone} = \text{yes} \iff \text{Calls} = \text{yes}, c_3: \text{GPS} = \text{yes} \rightarrow \text{Phone} = \text{yes}, c_4: \text{Phone} = \text{yes} \iff \text{Screen} = \text{yes}, c_5: \text{Media} = \text{yes} \rightarrow \text{Phone} = \text{yes}, c_6: \neg(\text{GPS}=\text{yes} \land \text{Basic} = \text{yes}), c_7: \text{Basic}=\text{yes} \rightarrow \text{Colour} = \text{no} \land \text{HighRes}=\text{no}, c_8: \text{Colour}=\text{yes} \rightarrow \text{Basic} = \text{no} \land \text{HighRes}=\text{no}, c_9: \text{HighRes}=\text{yes} \rightarrow \text{Colour} = \text{no} \land \text{Basic}=\text{no}, c_{10}: \text{Camera} = \text{yes} \rightarrow \text{HighRes} = \text{yes}, c_{11}: \text{Camera} = \text{yes} \rightarrow \text{Media} = \text{yes}, c_{12}: \text{MP3} = \text{yes} \rightarrow \text{Media} = \text{yes}, \ldots \} \]

\[ \text{S} = \{ \text{Phone}=\text{yes}, \text{Calls}=\text{yes}, \text{GPS}=\text{yes}, \text{Screen}=\text{yes}, \text{Media}=\text{yes}, \text{Basic}=\text{no}, \text{Colour}=\text{no}, \text{HighRes}=\text{yes}, \text{Camera}=\text{yes}, \text{MP3}=\text{no} \} \]

\[ \text{C}_R = \{ c_{13}: \text{GPS}=\text{yes}, c_{14}: \text{Camera} = \text{yes} \} \]
Why Configuration Technologies?

- Less Errors (e.g., incompatible components)
- Faster Response Times (e.g., immediate feedback for customer/user)
- Pre-informed Customers (knowledge about product/service)
- Increased Trust (explanations)
- Corporate Memory (standard level of service)

Weniger Fehler
Schnellere Antwortzeiten
Vorinformierte Kunden
Corporate Memory
Gesteigertes Vertrauen

[Felfernig and Kiener 2005]
[Felfernig et al. 2006]
Goal of this Presentation

Provide answers to the following questions:
• How to develop constraint-based systems efficiently?
• How to improve the quality of the underlying user interfaces?

Important Aspects:

1. Modeling the Knowledge Base
2. Debugging the Knowledge Base
3. Supporting the User
1. Modeling Knowledge Bases

Challenge: Knowledge representation languages have to be understandable (CSP, DCSP, GCSP) [Mittal and Falkenhainer 1990], [Fleischanderl et al. 1998]


Result (among others): Integration to commercial environment [Felfernig et al. 2006]
UML-based Modeling

- UML Profile (Configuration)
- "Well-formedness" Rules
- OCL Parser
- Context-sensitive Interface
- Automated Generation

context Computer inv:
((self.HDUnit->select(oclIsTypeOf(SCSIUnit))->size>0) and
(self_MB->select(oclIsTypeOf(MB1))->size>0)) implies false

Example: Generated Application

Debugging

1. Modeling Knowledge Bases

2. Debugging Knowledge Bases

3. Supporting the User

**Challenge:** Complex knowledge bases with frequent maintenance steps
[Barker et al. 1989], [Fleischanderl 2002], [Felfernig et al. 2007]

**Approach:** Automated Debugging of Knowledge Bases
[Felfernig et al. 2004, 2006]

**Result:**
Prototypical integration to commercial environment
[Felfernig et al. 2006]
Study @TUGraz: Are we able to identify faulty constraints in a knowledge base?

- **Course (2010):** Advanced Topics in AI
- **#Participants:** 14 Students (21% fem., 79% male.)
- **#Constraints:** 10 (binary)
- **#Variables:** 10 (domain size: 3)
- **Goal:** Conflicts & Diagnosis

Knowledge Bases: #Conflicts/Cardinality(Conflicts)

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>B1</th>
<th>B2</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/2</td>
<td>3/2</td>
<td>2/3</td>
<td>3/3</td>
<td>2/6</td>
<td>3/6</td>
</tr>
</tbody>
</table>

Scenario: Inconsistency in Testing

**Scenario:** \( \exists t_i \in T = \{t_1, \ldots, t_T\} \): inconsistent \( (C_{KB} \cup t_i) \)

**Conflict Set (CS):** \( CS \subseteq C_{KB} \) s.t. \( \exists t_i \in T \): inconsistent \( (CS \cup t_i) \)

**Minimal (CS):** \( \neg \exists CS' \) s.t. \( CS' \subset CS \)

**Explanation (\( \Delta \)):** \( \Delta \subseteq C_{KB} \): \((C_{KB} - \Delta) \cup t_i\) consistent \( \forall t_i \in T \)

**Minimal (\( \Delta \)):** \( \neg \exists \Delta' \) s.t. \( \Delta' \subset \Delta \)

Motivation • Modeling • Debugging

---

**Example:**
- \( c_1: x > y \)
- \( c_2: y = 2 \)
- \( t_1: x = 2 \)

**CS:** \( \{c_1, c_2\} \)

**\( \Delta_1 \):** \( \{c_1\} \)

---


---

**Should Be:** Test cases accepted
**But:** some not accepted
Scenario: Redundant Constraints

A constraint $c_i$ is called redundant iff $C_{KB} - \{c_i\} \models c_i$. If this condition is not fulfilled, $c_i$ is said to be non-redundant.

$C_{KB}$  
$C_1$ $C_2$ $C_3$ $\ldots$ $C_q$  

Should Be: Not redundant  
But: Redundancy exists

Scenario: Redundant ($C_{KB}$)  
Conflict Set (CS): $CS \subseteq C_{KB}$: $CS \cup \neg C_{KB}$ inconsistent  
Minimal (CS): $\neg \exists CS'$ s.t. $CS' \subset CS$


Configuration Benchmarks → Most knowledge bases are redundant!
ConfigWorks Debugging Environment

User Support


**Challenge:** Interaction support has potential for improvement, for example, „no solution could be found“ [Felfernig et al. 2004]

**Approach:** Explanations show the path to the solution [Felfernig et al. 2006, 2009, 2011]

**Result:**
- Integration in commercial applications
- Algorithmic improvements (FastDiag) [Felfernig et al. 2006, 2009, 2011]
Scenario: Inconsistent Requirements

\[ \Delta_1 \Delta_2 \ldots \Delta_d \]

**Should Be:** Consistent Requirements

**But:** Requirements inconsistent

**Scenario: Inconsistent** \((C_{KB} \cup C_R)\)

**Conflict Set (CS):** \(CS \subseteq C_R: CS \cup C_{KB}\) inconsistent

**Minimal (CS):** \(\neg \exists CS' \text{ s.t. } CS' \subset CS\)

**Explanation (\(\Delta\)):** \(\Delta \subseteq C_R: C_{KB} \cup (C_R - \Delta)\) consistent

**Minimal (\(\Delta\)):** \(\neg \exists \Delta' \text{ s.t. } \Delta' \subset \Delta\)


Determining Explanations ($C_R$)

- Minimal Cardinality Explanations [Reiter 1987]
- Explanations with high Probability [DeKleer 1990]
- Utility-based Explanations [Jannach and Liegl 2006]
- Corrective Explanations [O’Sullivan et al. 2005]
- Representative Explanations [O’Sullivan et al. 2007]
- kNN-based Explanations [Felfernig et al. 2009]
- Ensemble-based Explanations [Felfernig et al. 2011]

► Existing Approaches: good prediction quality but inefficient.
► FastDiag: Efficient determination of preferred explanations.

FastDiag: Preferred Explanations

$C_{KB} = \{c_1: 4\text{-wheel} = \text{yes} \rightarrow \text{type} = \text{xdrive},$  
$c_2: \text{skibag} = \text{yes} \rightarrow \text{type} \neq \text{city},$  
$c_3: \text{fuel} = 1.7 \rightarrow \text{type} = \text{city},$  
$c_4: \text{fuel} = 2.6 \rightarrow \text{type} \neq \text{xdrive},$  
$c_5: \text{type} = \text{combi} \rightarrow \text{skibag} = \text{yes},$  
$c_6: \text{type} = \text{limo} \rightarrow \text{pdc} = \text{yes}\}$

$C_R = \{c_7: \text{type} = \text{limo},$  
$c_8: \text{fuel} = 1.7,$  
$c_9: 4\text{-wheel} = \text{yes},$  
$c_{10}: \text{pdc} = \text{no},$  
$c_{11}: \text{color} = \text{black}\}$

Explanation:

$\Delta_1 = \{c_7, c_8\}$
FastDiag: Example

Principle: „divide and conquer“

Motivation • Modeling • Debugging • User Support

Conflict Sets:
- \{c_7, c_8\}
- \{c_7, c_9\}
- \{c_7, c_{10}\}
- \{c_8, c_9\}

Explanation:
\[ \Delta_3 = \{c_8, c_9, c_{10}\} \]

[Felfernig et al. 2011]
FastDiag: Runtimes (msec)

- Breadth First
- Best First
- Corrective Relax
- FastDiag

<table>
<thead>
<tr>
<th>SHORT</th>
<th>SOURCE</th>
<th># VAR</th>
<th># CON</th>
</tr>
</thead>
<tbody>
<tr>
<td>telecom</td>
<td><a href="http://www.variantum.com">www.variantum.com</a></td>
<td>242</td>
<td>84</td>
</tr>
<tr>
<td>bicycle</td>
<td><a href="http://www.itu.dk">www.itu.dk</a></td>
<td>35</td>
<td>78</td>
</tr>
<tr>
<td>car</td>
<td><a href="http://www.itu.dk">www.itu.dk</a></td>
<td>101</td>
<td>113</td>
</tr>
<tr>
<td>finserv</td>
<td><a href="http://www.hypo-alpe-adria.at">www.hypo-alpe-adria.at</a></td>
<td>25</td>
<td>15</td>
</tr>
</tbody>
</table>

Renault Knowledge Base (Configuration Benchmark)

[Felfernig et al. 2011]
# FastDiag: Precision

<table>
<thead>
<tr>
<th>Methode</th>
<th>N=1</th>
<th>N=3</th>
<th>N=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>breadth-first</td>
<td>0.12</td>
<td>0.39</td>
<td>0.62</td>
</tr>
<tr>
<td>utility-based</td>
<td>0.17</td>
<td>0.48</td>
<td>0.74</td>
</tr>
<tr>
<td>similarity-based</td>
<td>0.17</td>
<td>0.49</td>
<td>0.73</td>
</tr>
<tr>
<td>probability-based</td>
<td>0.15</td>
<td>0.47</td>
<td>0.74</td>
</tr>
<tr>
<td>ensemble-based</td>
<td>0.17</td>
<td>0.50</td>
<td>0.76</td>
</tr>
<tr>
<td>fastdiag</td>
<td>0.18</td>
<td>0.54</td>
<td>0.70</td>
</tr>
</tbody>
</table>


\[
\text{precision} = \frac{\#(\text{correct predictions})}{\#(\text{predictions})}
\]

avg. #diagnoses: 19.42  
std.dev.: 4.51
Research Issues

**Understandability**  Cognitive Psychology (CP)

**Complexity metrics**  Software Engineering + CP

**Refactoring rules**  Knowledge Engineering + CP

**Personalized Explanations**  Machine Learning (ML) + Recommender Systems (RS)

**Intelligent Navigation Support**  ML + RS

**Intelligent Maintenance Support**  ML + RS

<table>
<thead>
<tr>
<th>1st feature</th>
<th>2nd feature</th>
<th>3rd feature</th>
<th>4th feature</th>
<th>5th feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>user1</td>
<td>f2</td>
<td>f1</td>
<td>f4</td>
<td>f3</td>
</tr>
<tr>
<td>user2</td>
<td>f2</td>
<td>f1</td>
<td>f4</td>
<td>f3</td>
</tr>
<tr>
<td>user3</td>
<td>f1</td>
<td>f2</td>
<td>f3</td>
<td>f4</td>
</tr>
<tr>
<td>user4</td>
<td>f2</td>
<td>f1</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

User 4 would receive a recommendation for f4 (since the majority of nearest neighbors looked at f4).

Summary

• **Constraint-based systems are very popular** (e.g., CRM, Online Selling, Call Center, Project Management, Software Engineering)

• **Challenge**: large & complex knowledge bases

• Basis for **efficient development processes**:
  • domain-specific modeling languages
  • automated (personalized) debugging techniques

• **Efficient user interaction** on the basis of preferred explanations

• **Future research focus**: human-centered interaction with complex knowledge spaces.
Questions?
References (1)


References (2)


References (3)


References (4)


References (5)


Thank You!