International Workshop on Configuration
Toulouse, France, 2016

Towards Group-Based Configuration

Alexander Felfernig, Muesluem Atas, Trang Tran, Martin Stettinger

Institute for Software Technology
Graz University of Technology, Austria

{alexander.felfernig,muatas,ttrang,mstettinger}@ist.tugraz.at

Overview of the talk

- Introduction and motivation
- Group-based Configuration
- Resolving inconsistencies in group preferences
- Conclusion and future work

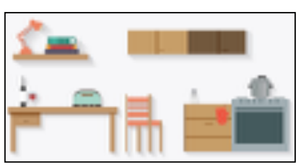
Configuration technologies

Single - user oriented

Problem:
sub-optimal
decisions

Telecommunication [1]
[Fleischanderl et al., 1998]

Financial services [1]
[Felfernig et al., 2007]



Furniture industry [1]
[Haag, 1998]

Group-based
Configuration

Group of users

Software Release Planning [1]
[Felfernig et al., 2012]

Holiday Planning
[Jameson et al., 2004]

[1]: www.freepik.com

Group-Based Configuration

Group-based Configuration Task

- **A group-based configuration task** can be defined on the basis of a Constraint Satisfaction Problem $CSP(V, D, C)$ [Tsang, 1993] where:
  - V is a set of variables.
  - D represents the corresponding domain definitions.
  - $C = PREF \cup CKB$ represents a set of constraints.
    - $PREF = \bigcup PREF_i$ is the union of customer preferences.
    - CKB represents a configuration knowledge base.

E. Tsang, Foundations of Constraint Satisfaction, Academic Press, London, 1993.

Group-based configuration task: Example

- A group-based configuration task from the *software release planning* domain:

$$V = \{req_1, ..., req_9\}$$

$$D = \{dom(req_1) = [1..3], .., dom(req_9) = [1..3]\}$$

$$PREF_1 = \{pref_{11} : req_1 = 1, pref_{12} : req_2 = 1, pref_{13} : req_3 = 1, pref_{14} : req_5 = 2, pref_{15} : req_8 = 3\}$$

$$PREF_2 = \{pref_{21} : req_3 = 1, pref_{22} : req_4 = 2, pref_{23} : req_6 = 3, pref_{24} : req_7 = 3\}$$

$$PREF_3 = \{pref_{31} : req_5 = 2, pref_{32} : req_6 = 3, pref_{33} : req_8 = 3, pref_{34} : req_9 = 2\}$$

$$CKB = \{c_1 : req_1 < req_5, c_2 : req_2 < req_8, c_3 : req_3 < req_6, c_4 : req_3 \neq req_4\}$$

Group-based Configuration

- **A group-based configuration** for a group-based configuration task is a complete set of assignments $CONF = \bigcup a_i : v_i = v_{ai}$ to the variables $v_i \in V$ such that $CONF \cup PREF \cup CKB$ is consistent.

- **Example:** A constraint solver could determine the following solution:
$CONF = \{a_1 : req_1 = 1, a_2 : req_2 = 1, a_3 : req_3 = 1, a4 : req_4 = 2,$
$a_5 : req_5 = 2, a_6 : req_6 = 3, a_7 : req_7 = 3, a8 : req_8 = 3, a9 : req_9 = 2\}$

Inconsistencies in Group Preferences

- Group-based configuration scenario: The preferences of individual users differ.
- In Release Planning scenarios:
  - Stakeholders have different preferences regarding the implementation of specific requirements.
  - A stakeholder has no preferences or does not understand the requirements in detail.

How to resolve inconsistencies?

[www.freepik.com]

How to resolve inconsistencies?

Showing inconsistent preferences to stakeholders → Minimal ***conflict sets*** are determined [Junker, 2004]

Stakeholders decide which changes should be performed → Conflict resolution is performed by users manually

Conflicts between requirements can be resolved **automatically** by calculating ***minimal diagnoses*** for minimal conflict sets.

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).

What is a conflict set?

- **A conflict set** $CS \subseteq \bigcup REQ_i$ is a minimal set of requirements such that *inconsistent(CS)*.
- A conflict set CS is **minimal** if there does not exist a conflict set CS' and $CS' \subset CS$.

Diagnosis

- Minimal conflict sets can be exploited for determining the corresponding **diagnoses** [Reiter, 1987].
- $\bigcup PREF_i \cup CKB$ is inconsistent, a minimal diagnosis represents a minimal set of requirements that has to be deleted from $\bigcup PREF_i$ such that a solution can be found for the remaining constraints.

R. Reiter, ‘A theory of diagnosis from first principles’, AI Journal, 23(1), 57–95, (1987).

Resolving inconsistencies: Example

- A group-based configuration task from *software release planning* domain:

$$V = \{req_1, ..., req_9\}$$

$$D = \{dom(req_1) = [1..3], .., dom(req_9) = [1..3]\}$$

$$CKB = \{c_1 : req_2 > req_1\ c_2 : req_2 < req_8, c_3 : req_3 < req_6,\ c_4 : req_3 \neq req_4\}$$

Stake-holder	req_1	req_2	req_3	req_4	req_5	req_6	req_7	req_8	req_9
1	$pref_{11}: req_1 = 2$	$pref_{12}: req_2 = 1$	$pref_{13}: req_3 = 1$		$pref_{14}: req_5 = 2$			$pref_{15}: req_8 = 3$	
2			$pref_{21}: req_3 = 2$	$pref_{22}: req_4 = 3$		$pref_{23}: req_6 = 3$	$pref_{24}: req_7 = 3$		
3					$pref_{31}: req_5 = 2$	$pref_{32}: req_6 = 3$		$pref_{33}: req_8 = 3$	$pref_{34}: req_9 = 2$

Resolving inconsistencies: Example

- A group-based configuration task from *software release planning* domain:

$$V = \{req_1, ..., req_9\}$$

$$D = \{dom(req_1) = [1..3], .., dom(req_9) = [1..3]\}$$

$$CKB = \{c_1 : req_2 > req_1, c_2 : req_2 < req_8, c_3 : req_3 < req_6,\ c_4 : req_3 \neq req_4\}$$

$CS_1 = \{pref_{11}, pref_{12}\}$ $CS_2 = \{pref_{13}, pref_{21}\}$

$pref_{11}$:	$pref_{12}$:	$pref_{13}$:	$pref_{21}$:
$req_1 = 2$	$req_2 = 1$	$req_3 = 1$	$req_3 = 2$

Corresponding set of alternative diagnoses (hitting sets):

$\Delta_1 = \{pref_{11}, pref_{13}\}$ $\Delta_3 = \{pref_{12}, pref_{13}\}$

$\Delta_2 = \{pref_{11}, pref_{21}\}$ $\Delta_4 = \{pref_{12}, pref_{21}\}$

Which diagnoses should be recommended first to the group?

Recommending diagnoses to the group

- Consider the impact of the different diagnoses on the preferences of stakeholders/users.
- Apply group decision heuristics [Masthoff, 2011] to figure out alternatives acceptable for the whole group.
  - ***Least Misery***
  - ***Average***
  - ***Most pleasure***

J. Masthoff, 'Group recommender systems', Recommender Systems Handbook, 677–702, (2011).

The impact of different diagnoses on stakeholders' preferences

stakeholder	$\Delta_1 = \{pref_{11}, pref_{13}\}$	$\Delta_2 = \{pref_{11}, pref_{21}\}$	$\Delta_3 = \{pref_{12}, pref_{13}\}$	$\Delta_4 = \{pref_{12}, pref_{21}\}$
1	2	1	2	1
2	0	1	0	1
3	0	0	0	0

- **Least misery / Average / Most pleasure:** $leastmisery(\Delta) = argmin_d \bigcup_{s \in users} pref_\delta(s, \Delta) = d$; $average(\Delta) = \frac{\sum_{s \in users} pref_\delta(s, \Delta)}{\#users}$; $mostpleasure(\Delta) = argmax_d \bigcup_{s \in users} pref_\delta(s, \Delta) = d$

Evaluation of the different diagnoses using group decision heuristics

stakeholder	$\Delta_1 = \{pref_{11}, pref_{13}\}$	$\Delta_2 = \{pref_{11}, pref_{21}\}$	$\Delta_3 = \{pref_{12}, pref_{13}\}$	$\Delta_4 = \{pref_{12}, pref_{21}\}$
least misery	2	1	2	1
average	0.67	0.67	0.67	0.67
most pleasure	0	0	0	0

Use ranking criteria "***less is better***" for selecting diagnoses

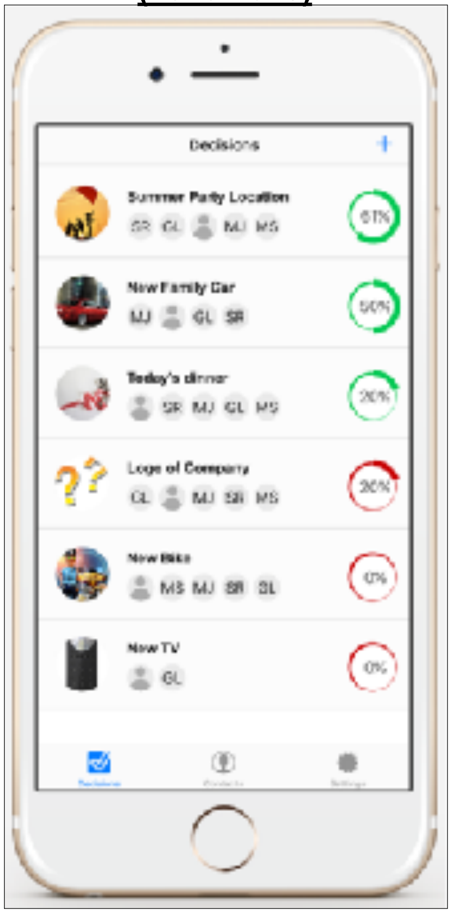
Conclusion

- Introduce a basic definition for Group-based Configuration task.
- How to deal with inconsistent preferences of group members.
- How to integrate decision heuristics into diagnosis selection processes.

Future work

WeWant
www.choicla.com
(2015-2018)

- *Consensus in Group Decision Making*:
  - Enrich user interfaces to allow basic negotiation mechanisms among users.
- *Fairness in Group Decision Making:*
  - The preferences of users discriminated in previous decisions should have a higher emphasis in the new decision.
- *Intelligent User Interfaces:*
  - *Support group-based configuration tasks in a distributed and asynchronous fashion.*
- ***Horizon (2017-2019):***
  - Intelligent Recommendation & Decision Technologies for Community-Driven Requirements Engineering.



CHOICLA group decision support environment [Stettinger, 2014]

M. Stettinger, ‘Choicla: Towards domain-independent decision support for groups of users’, in 8th ACM Conference on Recommender Systems, pp. 425–428, (2014).

Thank you for your listening!