Configuring Decision Tasks

Martin Stettinger[1] and **Alexander Felfernig**[1] and **Michael Jeran**[1] and **Gerald Ninaus** [1] and **Gerhard Leitner** [2] and **Stefan Reiterer** [3]

Abstract. In most cases, decision tasks are individual and different decision tasks require different combinations of features. Features can be, for instance, special preference visibilities during the decision process or specific heuristics that support the recommendation of decisions. To find the right features for a decision task it is essential to offer a corresponding configuration functionality. In this paper we illustrate how the design of a decision task can be represented as a configuration problem. The underlying configuration knowledge is already integrated in a tool called CHOICLA.

1 Introduction

Decisions have to be taken in different situations - for example a decision about the destination for the next holidays or a decision about which restaurant to choose for a dinner with friends. Decision scenarios can differ from each other in terms of their process design. Some decision scenarios rely on a preselected decision heuristic that defines the criteria for taking the decision, for example, a group decides to use majority voting for deciding about the next restaurant visit. Furthermore, the visibility of the preferences of other users is an important feature that can be configured by the creator of a decision task.

In this paper we show how the design of decision tasks (the underlying process) can be defined as a configuration problem. The major advantage of this approach is that making the process design of decision tasks configurable introduces the flexibility that is needed due to the heterogenity of decision problems. This way we are able to build a model that is flexible with regard to the implementation (generation) of problem-specific decision applications. The knowledge representations introduced in the following are included in the CHOICLA decision support environment (see www.choicla.com).

The remainder of this paper is organized as follows. In the next section (Section 2) we discuss features that are essential to the design of a decision task. In Section 3 we introduce dependencies that exist between features. In Section 4 we provide insights into group recommendation approaches integrated in the CHOICLA environment. We then discuss related and future work and thereafter conclude the paper.

2 Configuring a decision task

In the following we discuss different features that are relevant when designing (configuring) a decision task. On a formal level, we represent a *decision task configuration problem* as a constraint satisfaction problem [12] and [5] (CSP – see Definition 1).

Definition 1 (Constraint Satisfaction Problem). A CSP consists of (1) a set of finite-domain variables $X = \{x_1, x_2, ..., x_n\}$ and (2) a set of constraints $C = \{c_1, c_2, ..., c_m\}$. For each variable x_i out of X there exists a finite set D_i (domain of the variable) of possible assignments. Possible variable assignments can be limited via constraints. A complete assignment (every variable has a corresponding value) which is consistent with the constraints in C is denoted as a solution for a CSP.

For the purpose of better understandability we use a feature model notation to express variability properties of decision tasks. A feature model (FM) represents a set of possible features and relationships between them. Features are arranged hierarchically which is basically a tree structure with one root feature [2]. Within this tree structure the nodes are the features and the edges are the relationships (constraints). A more detailed discussion of different feature model representations can be found in [1], [2] and [3].

Six different types of constraints (relationships) are typically used for the construction of feature models ([1], [2]): *mandatory, optional, alternative, or, requires* and *excludes*. Feature models are representing configurable products which can be formalized in the form of a CSP. A feature f is *included* if the value is set to 1 - otherwise it is said to be *excluded*. We will exemplify this formalization on the basis of feature model depicted in Figure 1. Figure 1 shows a fragment of the CHOICLA feature model [4].
The CSP representation of the feature model depicted in Figure 1 is the following:

$V = \{f_1, f_2, ..., f_{21}\}$

$dom(f_1) = dom(f_2) = ... = dom(f_{21}) = \{0, 1\}$

$c_1 : f_1 \leftrightarrow f_2$
$c_2 : f_1 \leftrightarrow f_3$
$c_3 : f_1 \leftrightarrow f_4$
$c_4 : f_1 \leftrightarrow f_5$
$c_5 : f_6 \rightarrow f_1$
$c_6 : (f_7 \leftrightarrow (\neg f_8 \wedge f_3)) \wedge (f_8 \leftrightarrow (\neg f_7 \wedge f_3))$
$c_7 : (f_9 \leftrightarrow (\neg f_{10} \wedge \neg f_{11} \wedge f_4)) \wedge (f_{10} \leftrightarrow (\neg f_9 \wedge \neg f_{11} \wedge f_4)) \wedge (f_{11} \leftrightarrow (\neg f_9 \wedge \neg f_{10} \wedge f_4))$

[1] Graz, University of Technology, Austria, email: firstname.lastname@ist.tugraz.at
[2] Alpen Adria University, Austria, email: gerhard.leitner@aau.at
[3] SelectionArts Intelligent Decision Technologies GmbH, Austria, email: s.reiterer@selectionarts.com

[4] A more in-depth discussion of the CHOICLA decision support environment can be found in [18].

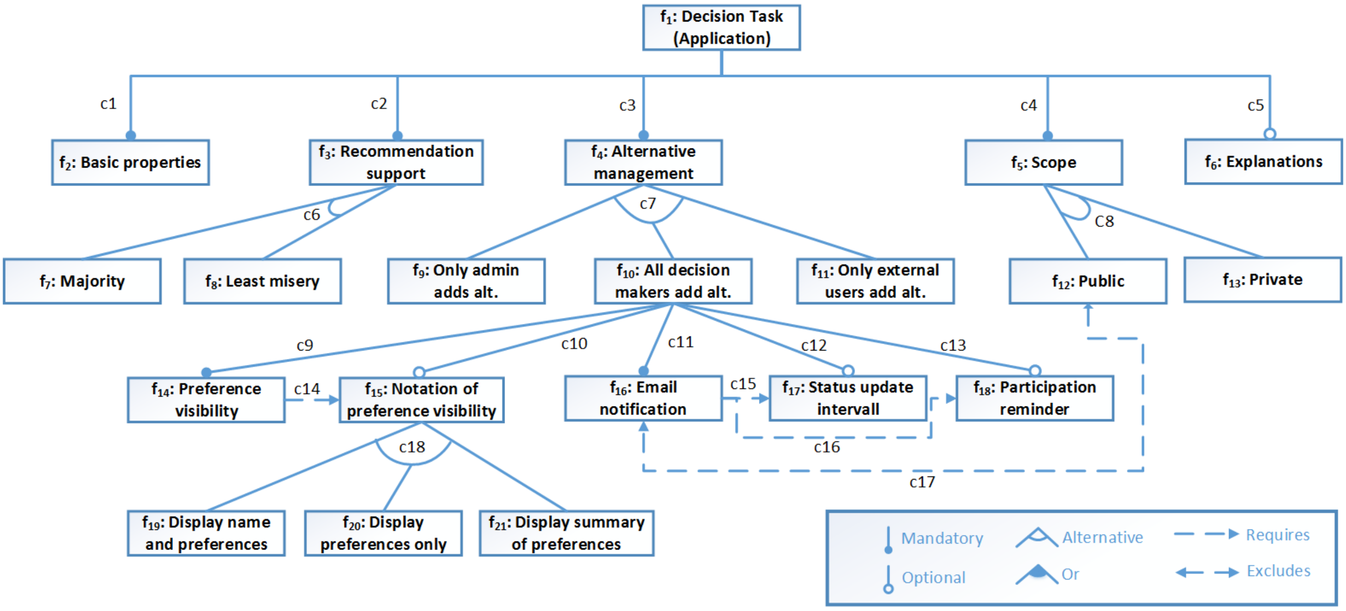

Figure 1. Fragment of the CHOICLA feature model. In this model, f_i are used as abbreviation for the individual features, for example, f_1 is the short notation for feature *Decision Task (Application)*.

$$c_8 : (f_{12} \leftrightarrow (\neg f_{13} \wedge f_5)) \wedge (f_{13} \leftrightarrow (\neg f_{12} \wedge f_5))$$
$$c_9 : f_{14} \leftrightarrow f_{10}$$
$$c_{10} : f_{15} \rightarrow f_{10}$$
$$c_{11} : f_{16} \leftrightarrow f_{10}$$
$$c_{12} : f_{17} \rightarrow f_{10}$$
$$c_{13} : f_{18} \rightarrow f_{10}$$
$$c_{14} : f_{14} \rightarrow f_{15}$$
$$c_{15} : f_{16} \rightarrow f_{17}$$
$$c_{16} : f_{16} \rightarrow f_{18}$$
$$c_{17} : \neg(f_{16} \wedge f_{12})$$
$$c_{18} : (f_{19} \leftrightarrow (\neg f_{20} \wedge \neg f_{21} \wedge f_{15})) \wedge (f_{20} \leftrightarrow (\neg f_{19} \wedge \neg f_{21} \wedge f_{15})) \wedge (f_{21} \leftrightarrow (\neg f_{19} \wedge \neg f_{20} \wedge f_{15}))$$

We will now discuss different basic properties of decision task configuration problems. In this context we explain the individual features and constraints depicted in Figure 1.

Basic properties. Each decision task is characterized by a name, a corresponding description, and a picture that represents the decision task (summarized in the feature *Basic Properties* for simplification purposes).

Management of alternatives. There are different possibilities to support alternative management within the scope of a decision task. *First*, only the creator of a decision task is allowed to add alternatives – this could be the case if a person is interested to know the opinions of his/her friends about a certain set of alternatives (e.g., alternative candidates for the next family car). Another related scenario are so-called "Micro-Polls" where the creator is only interested in knowing the preference distribution of a larger group of users. *Second*, in some scenarios it should be possible that all decision makers can add alternatives – a typical example of such a scenario is the group-based decision regarding a holiday destination or a hotel [10]. In this context, each user should be allowed to add relevant alternatives. An example scenario of the *third* case (only external users can add alternatives) is the support of group-based personnel decisions – in this context it should be possible that persons apply for a certain position (the application itself is interpreted as the addition of a new alternative to the decision task).

Scope. The scope of a decision task denotes the external visibility. The scope "private" allows only invited users to participate, i.e., the task is not visible for other users except those who have been invited. If the scope is "public", the decision task is visible to all users – this is typically the case in the context of so-called Micro-Polls. The selection of the scope has an impact on other features – related aspects will be discussed in Section 3.

Preference visibility. The visibility of individual preferences of the other participants involved in a decision process can have an impact on decision quality (see [6], [10], and [11]). There occur some decision scenarios where all participants should exactly know which person articulated a rating of an alternative. If, for example, a date for a business meeting is the topic of the decision task it is very essential to find a date where all division managers can attend the meeting and therefore it is important to know the individual preferences of the participants in that case. But there are of course decision scenarios where preference visibility can lead to disadvantages for some participants but still some kind of transparency of the preferences is helpful to come to the best decision. In such cases a summary of all given preferences of an alternative is a good way to support the participants best during the decision process. A summary prevents all participants from statistical inferences but still can help participants who are not sure about which rating to select.

Email notification. If this feature is set, emails can be used to exchange information about the current state of the decision process.

For example, the status update interval specifies in which intervals participants of a decision process receive a summary of the current status of the decision process. The active participation reminder is a feature which helps to trigger need for closure. If this feature is set, a maximum inactive time (without looking at the current status of the decision task) for the participants can be set. After this time is elapsed an email will be sent to the corresponding participants to encourage an active participation at the decision task.

Recommendation support. In context of group decision tasks another very essential aspect is the aggregation function (recommendation heuristic). Aggregation functions can help to foster consensus in a group decision process, furthermore, user studies show that these functions also help to increase the degree of the perceived decision quality (see, for example [6]). Preferences of individual users can be aggregated in many different ways and there exists no standard heuristic which fits for every decision scenario. To support groups of users in different scenarios the selection of recommendation heuristics is a necessary feature which has to be configured by the creator of a decision task. Some basic aggregation heuristics which can be used in such cases are described below. For an in-depth discussion of basic types of aggregation heuristics see, for example, the overview of Masthoff [14]. The example given in Table 1 represents the individual ratings of the participants for the defined alternatives. The results of applying the decision heuristics discussed below are depicted in Table 2.

restaurant	Martin	Dave	George	Ben
Clocktower	5	3	5	4
Häuserl im Wald	3	3	5	3
La Botte	5	3	3	3
El Gaucho	4	3	4	4

Table 1. Examples of user-specific ratings with regard to the available decision alternatives (restaurants).

Majority Voting (see Formula 1) determines the value (d) that a majority of the users selected as voting for a specific solution s where $eval(u, s)$ denotes the rating for solution s defined by user u. For example, the majority of votings for *Clocktower* is 5 (see Table 2).

$$MAJ(s) = maxarg_{(d \in \{1..5\})}(\#(\bigcup_{u \in Users} eval(u, s) = d)) \quad (1)$$

Least Misery (see Formula 2) returns the lowest voting for solution s as group recommendation. For example, the LMIS value for the s = *Clocktower* is 3.

$$LMIS(s) = min(\bigcup_{u \in Users} eval(u, s)) \quad (2)$$

Most Pleasure (see Formula 3) returns the highest voting for solution s as group recommendation. For example, the MPLS value for the s = *Clocktower* is 5.

$$MPLS(s) = max(\bigcup_{u \in Users} eval(u, s)) \quad (3)$$

Group Distance (see Formula 4) returns the value d as group recommendation which causes the lowest overall change of the individual user preferences. For example, the GDIS value for s = *Clocktower* is 5 (or, alternatively 4).

$$GDIS(s) = minarg_{(d \in \{1..5\})}(\sum_{u \in Users} |eval(u, s) - d|) \quad (4)$$

Finally, *Ensemble Voting* (see Formula 5) determines the majority of the results of the individual voting strategies H = {MAJ, LMIS, MPLS, GDIS}. For example, the ensemble-based majority voting for *Clocktower* is 5.

$$ENS(s) = maxarg_{(d \in \{1..5\})}(\#(\bigcup_{h \in H} eval(h, s) = d)) \quad (5)$$

solution	*MAJ*	*LMIS*	*MPLS*	*GDIS*	*ENS*
Clocktower	5	3	5	5	5
Häuserl im Wald	3	3	5	3	3
La Botte	3	3	5	3	3
El Gaucho	4	3	4	4	4

Table 2. Results of applying the aggregation functions to the user preferences shown in Table 1. MAJ = Majority Voting; LMIS = Least Misery; MPLS = Most Pleasure; GDIS = Lowest Group Distance; ENS = Ensemble Voting. This example is based on the preference information in Table 1.

Explanations. Explanations can play an important role in decision tasks since they are able to increase the trust of users in the outcome of a decision process [4]. When configuring a decision task in CHOICLA, explanations can be selected as a feature of the decision process. In the current version of CHOICLA, explanations are supported by simply allowing the creator of the decision process to include textual argumentations as to why a certain decision alternative has been selected as "the final decision". If this feature is selected, the administrator of a decision task has to enter some explanatory text, if not, the entering of such a text remains just an option.

3 Dependencies among features

We now discuss examples of constraints that restrict the combinations of features as shown in the feature model of Figure 1. The constraint-based representation of these constraints is shown in the CSP definition of the feature model given in Section 2.

Scope of a decision. If a decision task is public, there are restrictions regarding the support of message interchange (e.g., via email) and the visualization of the preferences of other users. In the case that a decision task is private, it is in both cases possible to choose. Preferences can (but must not) be made visible to other users and the type of possible message interchange can be specified. The differentiation between public and private decision tasks also has an impact on other system properties. For example, if a decision task is defined as private, the corresponding decision application can not be reused by other users, i.e., found as a result via the CHOICLA search interface.

Preference visibility. A dependency of type 'requires' exists between the feature *preference visibility* and the corresponding notation of visibility. Preference visibility denotes a functionality where the individual preferences of other users are made visible for the current user. The type of visualization can only be selected in the case that the *preference visibility* feature is has been selected by the designer of a decision task.

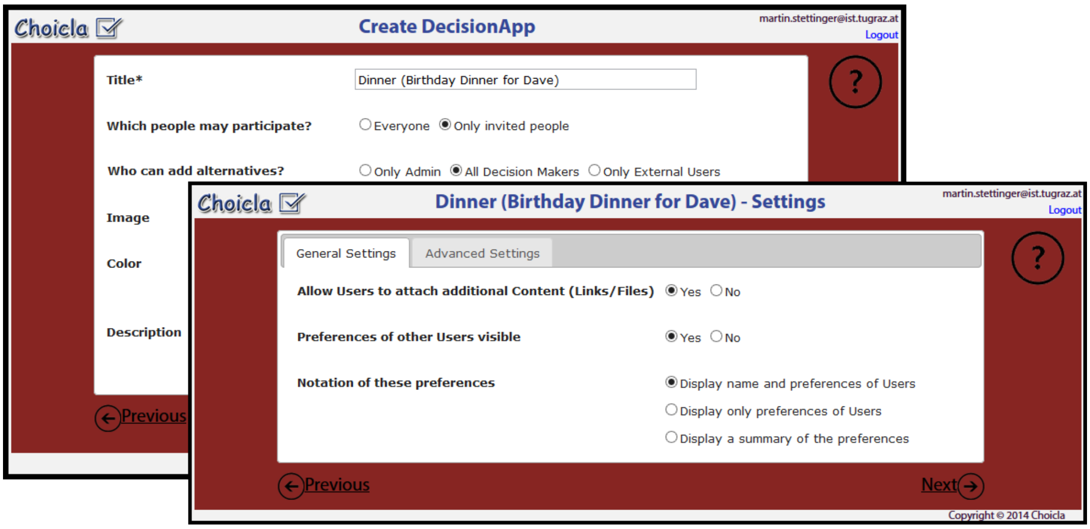

Figure 2. CHOICLA: definition of a decision task. Basic settings & further configurable features in case the decision makers are allowed to contribute own alternatives during the decision process.

Email notification. Similar to the visibility of preferences, the type of supported message exchange (e.g., via email) can only be specified in the case that the creator of the decision task decided to support email notifications. As already mentioned, email communication is only supported if the scope of the decision task is private.

These simple examples already show the need to manage decision task related variability in a structured fashion. Our knowledge representation approach allows for a product line oriented development of decision support functionalities and makes systems much more flexible for future requirements and corresponding extensions.

4 Configuring decision tasks in CHOICLA

In the following we give an example of how a decision task can be configured in the CHOICLA decision support environment (www.choicla.com). The application knowledge base of CHOICLA is currently rule-based. For reasons of easier maintenance and adaptability we apply reasoning and CSP for future versions of CHOICLA.

Parts of the user interface that supports a creator of a decision task are depicted in Figure 2. The possible parametrizations correspond to the features in the model of Figure 1. If, for example, a specific feature A depends on the inclusion of another feature B, this is taken into account in the user interface, i.e., such a feature (feature A) can only be selected, if the other feature (feature B) is also selected. In the example of Figure 2, the *scope* of the decision task is private (only invited users can participate), all decision makers are allowed to add alternatives, and for all participants of the decision process the preferences of other users are visible (names as well as preferences). Note that in the CHOICLA environment there are many additional features that can be selected within the scope of a decision task configuration process.
For understandability reasons we kept our working example simple and focused on aspects that give the reader an impression of the basic underlying configuration problem. The user interface for the inclusion of alternatives is depicted in Figure 3.
Figure 4 shows how the decision alternatives can be voted by the individual users of a decision task.

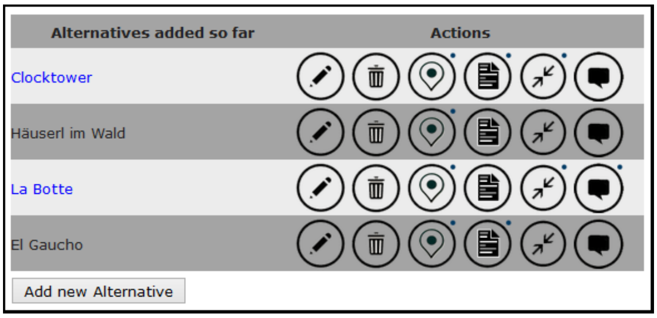

Figure 3. CHOICLA: user interface for addition of decision alternatives. The dots in the upper right corner of every symbol indicate whether there is an information in this category available or not. The meaning of the used symbols is (from left to right): *edit*, *delete*, *geographical information*, *files*, *links* and *comments*.

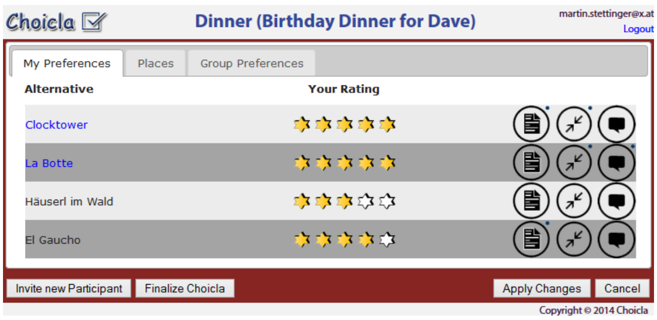

Figure 4. CHOICLA: user interface for individual voting of decision alternatives. Each alternative can be voted by a five-star scale. The tab *Places* shows the geographical distribution of the decision alternatives (if available). In tab *Group Preferences* the actual group recommendation as well as the individual preferences of the other users (if feature $f14$ is set) is presented to the users. The process where the "final decision" can be set is triggered by the button *Finalize Choicla*.

5 Related and Future Work

There exist a couple of online tools which support different types of decision scenarios. *The Decider*[5] is a tool that allows the creation of

[5] labs.riseup.net.

issues and decision alternatives – the corresponding recommendation is provided to users who are articulating their preferences regarding the given decision alternatives. Rodriguez et al. [17] introduce *Smartocracy* which is a decision support tool which supports the definition of tasks (issues or questions) and corresponding solutions. Solution selection (recommendation) is based on exploiting information from an underlying social network which is used to rank alternative solutions. *Dotmocracy*[6] is a method for collecting and visualizing the preferences of a large group of users. It is related to the idea of participatory decision making – it's major outcome is a graph type visualization of the group-immanent preferences. Doodle[7] focuses on the aspect of coordinating appointments – similarly, VERN [19] is a tool that supports the identification of meeting times based on the idea of unconstrained democracy where individuals are enabled to freely propose alternative dates themselves. Compared to CHOICLA these tools are not able to customize their decision processes depending on the application domain and are also focused on specific tasks. Furthermore, no concepts are provided which help to improve the overall quality of group decisions, for example, in terms of integrating explanations, recommendations for groups, and consistency management for user preferences.

The support of group decision processes on the basis of recommendation technologies is a new and upcoming field of research (see, e.g., Masthoff et al. [14]). The application of group recommendation technologies is still restricted to specific domains such as interactive television [13], e-tourism [9, 15], software requirements engineering [6], and ambient intelligence [16].

Future Work. Our future work will focus on the analysis of further application domains for the CHOICLA technologies. Our vision is to make the design (implementation) of group decision tasks as simple as possible. The resulting decision task should be easy to handle for users and make group decisions in general more efficient. Within the scope of our work we will also focus on the analysis of decision phenomena within the scope of group decision processes. Phenomena such as decoy effects [7] and anchoring effects [8] are well known for single-user cases but are not investigated in group-based decision scenarios. Finally, we will also focus on the development of further group recommendation heuristics. In this context, our major goal is to make the CHOICLA datasets available to the research community in an anonymized fashion for experimentation purposes.

6 Conclusions

In this paper we have shown how to represent the design of decision tasks as a configuration problem. In this context, we gave a short introduction to the CHOICLA group decision environment which supports the flexible design and execution of different types of group decision tasks. Compared to existing group decision support approaches, CHOICLA provides an end user modelling environment which supports an easy development and execution of group decision tasks.

ACKNOWLEDGEMENTS

The work presented in this paper has been conducted in the research project PEOPLEVIEWS funded by the Austrian Research Promotion Agency (843492).

[6] dotmocracy.org.
[7] doodle.com.

REFERENCES

[1] Don Batory, 'Feature models, grammars, and propositional formulas', in *Proceedings of the 9th International Conference on Software Product Lines*, SPLC'05, pp. 7–20, Berlin, Heidelberg, (2005). Springer-Verlag.
[2] David Benavides, Sergio Segura, and Antonio Ruiz-Corts, 'Automated analysis of feature models 20 years later: A literature review', *Information Systems*, **35**(6), 615 – 636, (2010).
[3] A. Felfernig, D. Benavides, J. Galindo, and F. Reinfrank, 'Towards Anomaly Explanation in Feature Models', *Workshop on Configuration, Vienna, Austria*, 117–124, (2013).
[4] A. Felfernig, B. Gula, and E. Teppan, 'Knowledge-based Recommender Technologies for Marketing and Sales', *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, **21**(2), 1–22, (2006).
[5] Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, *Knowledge-based Configuration – From Research to Business Cases*, Elsevier, 2014.
[6] Alexander Felfernig, Christoph Zehentner, Gerald Ninaus, Harald Grabner, Walid Maalej, Dennis Pagano, Leopold Weninger, and Florian Reinfrank, 'Group decision support for requirements negotiation', in *Advances in User Modeling*, eds., Liliana Ardissono and Tsvi Kuflik, volume 7138 of *Lecture Notes in Computer Science*, 105–116, Springer Berlin Heidelberg, (2012).
[7] J. Huber, J. Payne, and C. Puto, 'Adding Asymmetrically Dominated Alternatives: Violations of Regularity and the Similarity Hypothesis', *The Journal of Consumer Research*, **9**(1), 90–98, (1982).
[8] K. Jacowitz and D. Kahneman, 'Measures of Anchoring in Estimation Tasks', *Personality and Social Psychology Bulletin*, **21**(1), 1161–1166, (1995).
[9] A. Jameson, S. Baldes, and T. Kleinbauer, 'Two methods for enhancing mutual awareness in a group recommender system', in *ACM Intl. Working Conference on Advanced Visual Interfaces*, pp. 48–54, Gallipoli, Italy, (2004).
[10] Anthony Jameson, 'More than the sum of its members: challenges for group recommender systems', in *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pp. 48–54, New York, NY, USA, (2004). ACM.
[11] Anthony Jameson and Barry Smyth, 'Recommendation to groups', in *The Adaptive Web*, eds., Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, volume 4321 of *Lecture Notes in Computer Science*, 596–627, Springer Berlin Heidelberg, (2007).
[12] Alan Mackworth, 'Consistency in networks of relations', *Artificial Intelligence*, **8**(1), 99–118, (1977). Reprinted in Readings in Artificial Intelligence.
[13] J. Masthoff, 'Group modeling: Selecting a sequence of television items to suit a group of viewers', *User Modeling and User-Adapted Interaction (UMUAI)*, **14**(1), 37–85, (2004).
[14] J. Masthoff, 'Group Recommender Systems: Combining Individual Models', *Recommender Systems Handbook*, 677–702, (2011).
[15] K. McCarthy, M. Salamo, L. Coyle, L. McGinty, B. Smyth, and P. Nixon, 'Group recommender systems: a critiquing based approach', in *2006 International Conference on Intelligent User Interfaces (IUI 2006)*, pp. 282–284, Sydney, Australia, (2006). ACM.
[16] I. Perez, F. Cabrerizo, and E. Herrera-Viedma, 'A Mobile Decision Support System for Dynamic Group Decision-Making Problems', *IEEE Transactions on Systems, Man, and Cybernetics*, **40**(6), 1244–1256, (2010).
[17] M. Rodriguez, D. Steinbock, J. Watkins, C. Gershenson, J. Bollen, V. Grey, and B. deGraf, 'Smartocracy: Social networks for collective decision making', in *HICSS 2007*, p. 90, Waikoloa, Big Island, HI, USA, (2007). IEEE.
[18] Martin Stettinger, Gerald Ninaus, Michael Jeran, Florian Reinfrank, and Stefan Reiterer, 'We-decide: A decision support environment for groups of users', in *Recent Trends in Applied Artificial Intelligence*, eds., Moonis Ali, Tibor Bosse, KoenV. Hindriks, Mark Hoogendoorn, CatholijnM. Jonker, and Jan Treur, volume 7906 of *Lecture Notes in Computer Science*, 382–391, Springer Berlin Heidelberg, (2013).
[19] S. Yardi, B. Hill, and S. Chan, 'VERN: Facilitating Democratic group Decision Making Online', in *International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2005)*, pp. 116–119, Sanibel Island, Florida, USA, (2005). ACM.