International Journal of Systems Science
Vol. 00, No. 00, 00 Month 20xx, 1–16

RESEARCH ARTICLE

JSMAA: open source software for SMAA computations

Tommi Tervonen[†]
Econometric Institute, Erasmus University Rotterdam
PO Box 1738, 3000DR Rotterdam, The Netherlands
(*Received 00 Month 20xx; final version received 00 Month 20xx*)

Most software for Multi-Criteria Decision Analysis (MCDA) implement a small set of compatible methods as a closed monolithic program. With such software tools, the decision models have to be input by hand. In some applications, however, the model can be generated using external information sources, and thus it would be beneficial if the MCDA software could integrate in the comprehensive information infrastructure. This paper motivates for the need of model generation in the methodological context of Stochastic Multicriteria Acceptability Analysis (SMAA), and describes the JSMAA software that implements SMAA-2, SMAA-O and SMAA-TRI methods. JSMAA is open source and divided in separate graphical user interface and library components, enabling its use in systems with a model generation subsystem.
Keywords: Stochastic Multicriteria Acceptability Analysis (SMAA); Open Source Software; JSMAA; Decision Support Systems (DSS); Multi-Criteria Decision Analysis (MCDA)

1. Introduction

Multiple Criteria Decision Analysis/Aiding (MCDA, also referred to as -Making, MCDM) considers decision problems in which several decision alternatives are evaluated in terms of multiple criteria. The evaluation is done in order to either choose the best/small subset of alternatives, to rank them, or to sort them into ordered categories (Roy 1996). Although the amount of published MCDA applications has increased substantially in the past 15 years (Wallenius et al. 2008), the impact of decision support systems (including implementation of MCDA) in managerial practice has not (Keen and Sol 2008). For a comprehensive list of decision support systems, see Weistroffer et al. (2005). There are various possible reasons for the lack of impact, including the following:

(1) The scientific MCDA community has produced a vast amount of *useful* analytical methods, but a majority of them are not supported by software providing a *usable* interface between the user and the decision support technology, let alone be *used* in real-life decision processes (our definition of usefulness, usability, and usage is similar to Keen and Sol 2008). Most of the MCDA software tools implement a single method or a small set of similar methods (French and Xu 2005, Belton and Hodgkin 1999). The software is often developed in an academic environment, is closed source, and requires a license for full use (Weistroffer et al. 2005). So, the licensing model is often commercial, but the development status "experimental" and features provided by the software are limited.
(2) The reported real-life use of MCDA methods appears mainly in disciplines where models can be constructed and calculated manually or with a general purpose software, e.g. in location (cf. Nickel et al. 2005) or in financial decision making (cf. Spronk et al. 2005). The limited practical application of MCDA in new disciplines can be due to difficulty

[†]E-mail: tervonen@ese.eur.nl

of integrating existing MCDA software with rest of the decision support technology (e.g. simulation models (Law and Kelton 1999) or geographic information systems (Malczewski 1999)). Even when artificial intelligence techniques are applied, the decision support system is often monolithic (Siskos and Spyridakos 1999).

The use cases and software of MCDA have previously been considered by French and Xu (2005) and Belton and Hodgkin (1999). They defined multi-criteria problems as one-off decisions where the decision making process starts with a construction of a model and terminates with an evaluation of the decision (the process can be iterative). Although one-off decision problems occur often in practice, there are also repeated decision contexts, such as ranking of suppliers or universities, that can be supported through a MCDA subsystem with a model generation module. We distinguish the terms *model generation* and *model construction*. Construction is defined as a manual process, whereas generation refers to an automated one. This paper motivates for the need of more modular MCDA software components and describes one such implementation that enables model generation. The need for such software is considered in the methodological context of Stochastic Multicriteria Acceptability Analysis (SMAA).

SMAA is a family of MCDA methods for all MCDA problem statements (Tervonen and Figueira 2008). The methods are based on inverse parameter space analysis through Monte Carlo simulation. The different SMAA methods allow tackling problems with uncertain, imprecise, and (partially) incomplete information about the preferences, the technical parameters, and the criteria measurements. Incomplete information means that the value is missing, whereas imprecise information means that the value is present but not with the required precision. Uncertainty takes into account the observer, who is assumed to give complete and precise information, but is unreliable itself (Smets 1991). From the technical point of view, the SMAA methods handle all three types of ignorance about the parameter values in a similar way through probability distributions or as ordinal information. In practice the SMAA methods cannot be calculated manually due to the use of simulation; a software implementation is required. This paper describes the first stable SMAA software, JSMAA, that currently implements the SMAA-2, SMAA-O and SMAA-TRI methods. JSMAA is developed in Java and licensed under open source (for more information about the open source development model, see e.g. Lakhani and von Hippel 2003) GNU General Public License v3, and can help to mitigate the problems described before by enabling model generation.

Section 2 introduces the reader to the SMAA methods implemented in JSMAA. An example application of drug benefit-risk analysis with model generation is presented in Section 3. The JSMAA software is presented in Section 4. Section 5 concludes with discussion of alternative ways of designing modular MCDA software.

2. SMAA methodolody

SMAA methods consider a discrete decision-making problem where of a set of m alternatives $X = \{x_1, \ldots, x_i, \ldots, x_m\}$ are evaluated on the basis of a set of n criteria $\{g_1, \ldots, g_j, \ldots, g_n\}$. The evaluation of alternative x_i on criterion g_j is denoted by $g_j(x_i)$. The core idea in SMAA methods is to compute descriptive measures based on multidimensional integrals over stochastic parameter spaces. We will here describe the SMAA-2 and SMAA-TRI methods, and SMAA-O for handling ordinal criteria measurements with value theory. For full description of the methodology, see Tervonen and Figueira (2008), and for the actual algorithms, Tervonen and Lahdelma (2007).

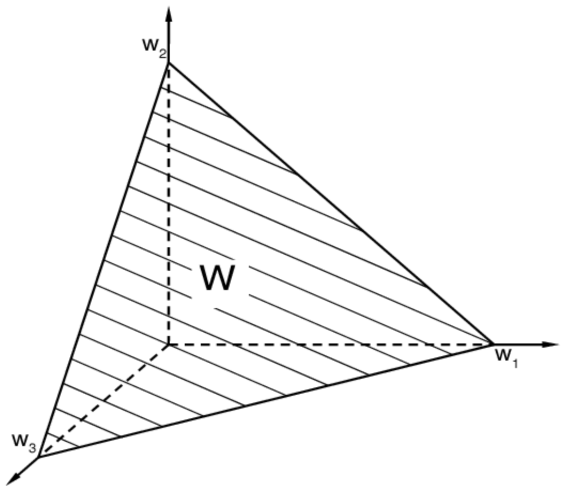

Figure 1. The feasible weight space of a 3-criteria problem.

2.1. *SMAA-2*

The SMAA-2 (Lahdelma and Salminen 2001) is for ranking the m alternatives. It considers a preference structure representable with an individual weight vector w and a real-valued utility or value function $u(x_i, w)$, $i = 1, \ldots, m$. The most commonly used value function is the linear one:

$$u(x_i, w) = \sum_{j=1}^{n} u_j(g_j(x_i))w_j, \tag{1}$$

where $u_j(\cdot)$ are the partial value functions (Belton and Stewart 2002). The weights are considered to be non-negative and normalized, therefore defining the feasible weight space:

$$W = \left\{ w \in R^n : w_j \geq 0 \text{ and } \sum_{j=1}^{n} w_j = 1 \right\}. \tag{2}$$

The feasible weight space of a 3-criteria problem with no preference information is illustrated in Figure 1.

The SMAA methods are developed for situations where neither criteria values nor weights or other parameters of the model are precisely known (Tervonen and Figueira 2008). Uncertain or imprecise criteria values are represented by stochastic variables ξ_{ij} (corresponding to the deterministic evaluations $g_j(x_i)$) with assumed or estimated joint probability function distribution and density function $f_\chi(\xi)$ in the space $\chi \subseteq R^{m \times n}$. Similarly, the Decision Maker (DM)s unknown or partially known preferences are represented by a weight distribution with a joint density function $f_W(w)$ in the feasible weight space W. Total lack of preference information on the weights is represented by a uniform weight distribution in W, that is:

$$f_W(w) = 1/\mathrm{vol}(W). \tag{3}$$

The SMAA-2 introduces three such measures: the rank acceptability index, the central weight vector, and the confidence factor. For this purpose, a ranking function is defined as follows:

$$rank(i, \xi, w) = 1 + \sum_{k=1}^{m} \rho\Big(u(\xi_k, w) > u(\xi_i, w)\Big), \tag{4}$$

where $\rho(true) = 1$ and $\rho(false) = 0$. Let us also define the sets of favourable rank weights $W_i^r(\xi)$ as follows,

$$W_i^r(\xi) = \{w \in W : rank(i, \xi, w) = r\}. \tag{5}$$

2.1.1. *Rank acceptability index*

The rank acceptability index b_i^r describes the share of parameter values granting alternative x_i rank r. It is computed as a multidimensional integral over the criteria distributions and the favourable rank weights as follows,

$$b_i^r = \int_{\xi \in \chi} f_\chi(\xi) \int_{w \in W_i^r(\xi)} f_W(w)\, dw\, d\xi. \tag{6}$$

The most acceptable (best) alternatives are those with high acceptabilities for the best ranks. Evidently, the rank acceptability indices are within the range [0,1], where 0 indicates that the alternative will never obtain a given rank and 1 indicates that it will obtain the given rank always with any choice of weights.

2.1.2. *Central weight vector*

The central weight vector w_i^c is defined as the expected center of gravity of the favourable weight space. It is computed as a multidimensional integral over the criteria and weight distributions as

$$w_i^c = \int_{\xi \in \chi} f_\chi(\xi) \int_{w \in W_i^1(\xi)} f_W(w) w\, dw\, d\xi / b_i^1. \tag{7}$$

The central weight vector describes the preferences of a typical DM supporting this alternative with the assumed preference model. By presenting the central weight vectors to the DMs, an inverse approach for decision support can be applied: instead of eliciting preferences and building a solution to the problem, the DMs can learn what kind of preferences lead into which alternatives without providing any preference information.

2.1.3. *Confidence factor*

The confidence factor p_i^c is defined as the probability for an alternative to be the preferred one with the preferences expressed by its central weight vector. It is computed as a multidimensional integral over the criteria distributions as follows,

$$p_i^c = \int_{\xi \in \chi : u(\xi_i, w_i^c) \geq u(\xi_k, w_i^c)} f_\chi(\xi)\, d\xi. \tag{8}$$

Confidence factors can be calculated similarly for any given weight vectors. The confidence factors measure whether the criteria measurements are accurate enough to discern the efficient alternatives. If the problem formulation is to choose an alternative to implement, the ones with low confidence factors should not be chosen. If they are deemed as attractive ones, more accurate criteria data should be collected in order to make a reliable decision.

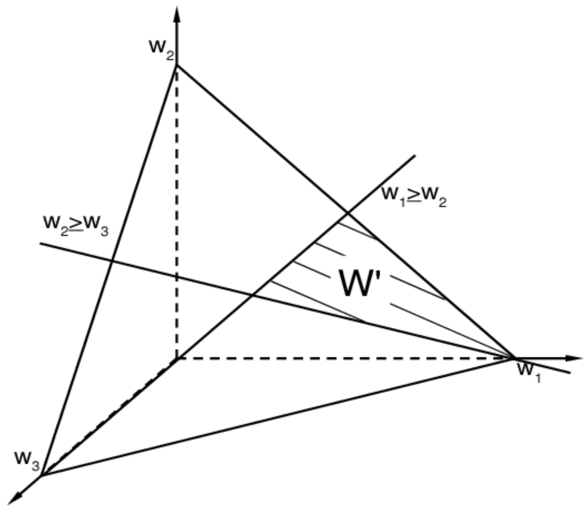

Figure 2. The feasible weight space of a 3-criteria problem with complete ranking of the weights.

2.1.4. Preference information

In most decision-making problems it is possible to elicit some, though probably imprecise and uncertain, preference information from the DMs. Although SMAA allows preference information to be represented with an arbitrary density function, usually it is easier to elicit the preferences as constraints for the weight space. Then the density function is defined with a uniform distribution in the restricted weight space W' as

$$f_{W'}(w) = \begin{cases} 1/\mathrm{vol}(W'), & \text{if } w \in W', \\ 0, & \text{if } w \in W \setminus W'. \end{cases} \tag{9}$$

Figure 2 illustrates the feasible weight space of a 3-criteria problem with complete ranking of the weights.

2.1.5. Ordinal measurements

SMAA-O (Lahdelma et al. 2003) extends SMAA to consider ordinal criteria measurements, meaning that the DMs have ranked the alternatives according to each (ordinal) criterion. In SMAA-O, the ordinal information is mapped to cardinal without forcing any specific mapping. This means that nothing is assumed about the cardinal rank values in the piecewise linear mapping.

The ordinal criteria are measured by assigning for each alternative a rank level number $r_j = 1, \ldots, j^{max}$, where 1 is the best and j^{max} the worst rank level. Alternatives considered equally good are placed on the same rank level and the rank levels are numbered consecutively. On an ordinal scale, the scale intervals do not contain any information, and should be therefore treated as such without imposing any extra assumptions. However, some mapping can be assumed to underlie the ordinal information. In SMAA-O, all mappings that are consistent with the ordinal information are simulated numerically during the Monte Carlo iterations. This means generating random cardinal values for the corresponding ordinal criteria measurements in a way that preserves the ordinal rank information. Figure 3 illustrates a sample mapping generated in this way.

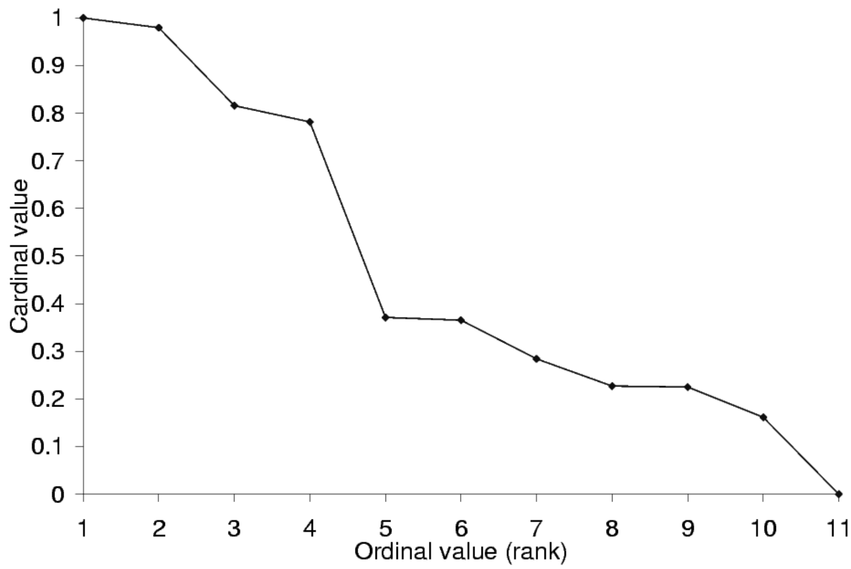

Figure 3. An sample ordinal to cardinal mapping of SMAA-O.

2.2. *SMAA-TRI*

ELECTRE TRI (Yu 1992) is a method for sorting problem statements, where the alternatives are to be assigned to pre-defined and ordered categories. SMAA-TRI (Tervonen et al. 2009) extends it to allow uncertainty on the parameter values. Let us denote by $C = \{C_1, \ldots, C_h, \ldots, C_k\}$ the set of categories in ascending preference order (C_1 is the "worst" category). These categories are defined by upper and lower profiles, that are computationally equivalent to alternatives. The profiles are denoted as $p_1, \ldots, p_h, \ldots, p_{k-1}$. Profile p_h is the upper limit of category C_h and the lower limit of category C_{h+1}. Notice that the profiles are strictly ordered, that is they have to satisfy

$$p_1 \,\Delta\, p_2 \,\Delta\, \ldots \,\Delta\, p_{k-2} \,\Delta\, p_{k-1}, \tag{10}$$

where Δ is the dominance relation ($p_1 \Delta p_2$ means that p_2 dominates p_1). This dominance relation needs to be interpreted in a wide sense, because the domination depends not only on the values of components of the two profiles, but also on the values of thresholds. The outranking model applied in SMAA-TRI applies three thresholds: preference threshold that describes the minimum difference for an alternative to be at least as good as another with respect to a single criterion, the indifference threshold for maximum difference considered to be insignificant, and the veto threshold for minimum difference that is so large, that no matter what are the values for other criteria, the alternative cannot be better than the other one. The actual preference model and the assignment procedure are described in Tervonen et al. (2009). For the assignment procedure an additional technical parameter, the lambda cutting level, has to be defined.

The input for ELECTRE TRI in SMAA-TRI is denoted as follows:

(1) Uncertain or imprecise profiles are represented by stochastic variables ϕ_{hj} with a joint density function $f_\Phi(\phi)$ in the space $\Phi \subseteq R^{(k-1)\times n}$. The joint density function must be such that all possible profile combinations satisfy (10). Usually the category profiles are defined to be independently distributed, and in this case the distributions must not overlap. For example, if the profile values for a criterion are normal distributed, the distributions must have tails truncated as shown by the vertical lines in Figure 4.
(2) The lambda cutting level is represented by a stochastic variable Λ with a density function

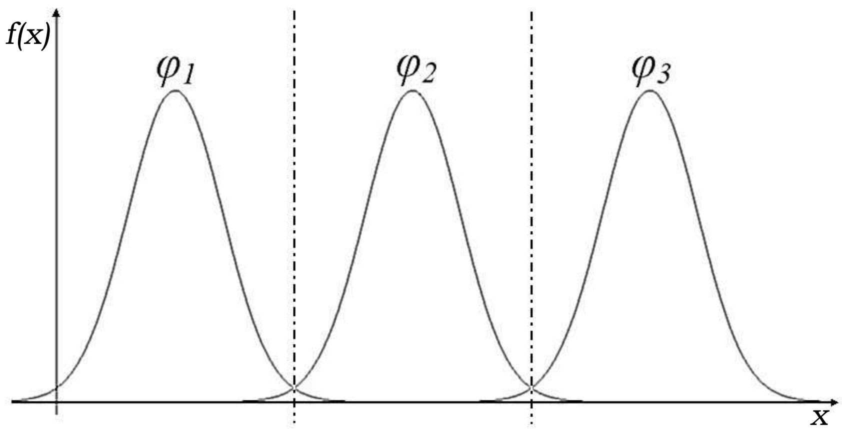

Figure 4. Probability distribution functions for three normal distributed profile values (for a single criterion) in SMAA-TRI. The vertical lines show where the tails of the distributions must be truncated.

$f_L(\Lambda)$ defined within the valid range [0.5,1].

(3) The weights and criteria measurements are represented as in SMAA-2.

(4) The data and other parameters of ELECTRE TRI are represented by the set $T = \{M, q, p, v\}$, where q, p and v are the model indifference, preference and veto thresholds. These components are considered to have deterministic values.

SMAA-TRI produces category acceptability indices for all pairs of alternatives and categories. The category acceptability index π_i^h describes the share of possible parameter values that have an alternative x_i assigned to category C_h. Let us define a *categorization function* that evaluates the category index h to which an alternative x_i is assigned by ELECTRE TRI:

$$h = K(i, \Lambda, \phi, w, T), \tag{11}$$

and a category membership function

$$m_i^h(\lambda, \phi, w, T) = \begin{cases} 1, \text{ if } K(i, \Lambda, \phi, w, T) = h, \\ 0, \text{ otherwise,} \end{cases} \tag{12}$$

which is applied in computing the category acceptability index numerically as a multi-dimensional integral over the finite parameter spaces as

$$\pi_i^h = \int_{0.5}^{1} f_L(\Lambda) \int_{\phi \in \Phi} f_\Phi(\phi) \int_{w \in W} f_W(w) m_i^h(\Lambda, \phi, w, T)\, dw\, d\phi\, d\Lambda. \tag{13}$$

The category acceptability index measures the stability of the assignment, and it can be interpreted as a probability for membership in the category. If the parameters are stable, the category acceptability indices for each alternative should be 1 for one category, and 0 for the others.

2.3. *Model generation with SMAA*

The SMAA-2 *ranking* problem consists of a set of alternatives that are evaluated based on a set of criteria. These are aggregated with the preference model to construct a complete pre-order of the alternatives. The partial value functions are scaled with weights that are defined as importances of scale swing from the worst to the best criterion performances (Belton and Stewart 2002). Therefore the weights (and even the partial value functions if not assumed linear) can only be defined when all the decision alternatives are known, and in cases where the alternatives are not known beforehand (e.g. they are generated from a database) the model generation cannot be completely automated. For example, consider a repeated evaluation problem of ranking suppliers performances: when another supplier with a criterion performance outside the interval hull of performances of the existing ones becomes available, the weights have to be re-elicited based on the modified scale swings. The following section illustrates model generation with an application of SMAA-2 in drug benefit-risk analysis.

3. Application: drug benefit-risk analysis

Drug benefit-risk analysis is done daily by health care professionals, such as regulators, practicing physicians, and employees of insurance companies, to evaluate the safety and efficacy of different medical compounds. The benefit is often evaluated as the efficacy of a compound over an active comparator or placebo. The risks can be defined as increase in the amounts of Adverse Drug Reactions (ADRs). Tervonen et al. (2011) proposed to apply SMAA-2 in drug benefit-risk assessment. In their example application, the model was constructed for a set of second generation antidepressants. The criteria considered were efficacy defined as the treatment response (benefit criterion) and the most common adverse drug events (risk criteria). All criteria were measured in the original clinical trial as incidence rates.

The observed incidences were considered to be realizations from binomially distributed variables, and by assuming their independence, the criteria performances were modeled with Beta distributions estimated through a Bayesian approach. Linear partial value functions were used, and their ranged defined as interval hulls (i.e. the smallest interval containing all) of the Beta distributed measurements for that criterion. Then, an expert in the field was asked to provide ordinal weight information (ranking of the scale swings) by considering two scenarios: mild and severe depression. In addition a preference-free model was used to quantify preferences supporting each treatment to be the preferred one.

The information flow of this case is presented in Figure 5. Note that when linear partial value functions are assumed, only two inputs are required from the DM: selection of the relevant outcomes (=criteria), and ranking of the scale swings. As can be seen from Figure 5, preference elicitation can only be made after partial value functions have been constructed, and therefore a usable decision support system for clinical trial evaluation through MCDA must include a model generation subsystem. Such an approach has been taken in the ADDIS software that uses JSMAA for the computation of its benefit-risk models (van Valkenhoef et al. 2012).

4. JSMAA

JSMAA (`www.smaa.fi`) supercedes CSMAA that was developed partly to support application of SMAA-TRI in nanomaterial risk assessment (Tervonen et al. 2009). The development continued to enable application of SMAA-2 in drug benefit-risk assessment as described in the previous section. JSMAA was always developed independently of any application and thus enables the use of SMAA methodology in other contexts well. Currently (as of version 0.8.4) JSMAA implements SMAA-TRI for sorting problems and SMAA-2 for ranking with multi-attribute value/utility theory. JSMAA computations are made with 10 000 Monte Carlo iterations, leading to a $\pm 1\%$ accuracy (with 95% confidence) for the computed rank/category acceptability indices (Tervonen and Lahdelma 2007).

The standard decision process of using JSMAA is illustrated in Figure 6. The process begins with the DM defining the alternatives and then the criteria, or vice versa if a value-focuced decision paradigm (Keeney 1996) is more suitable. Then (imprecise) measurements must be input.

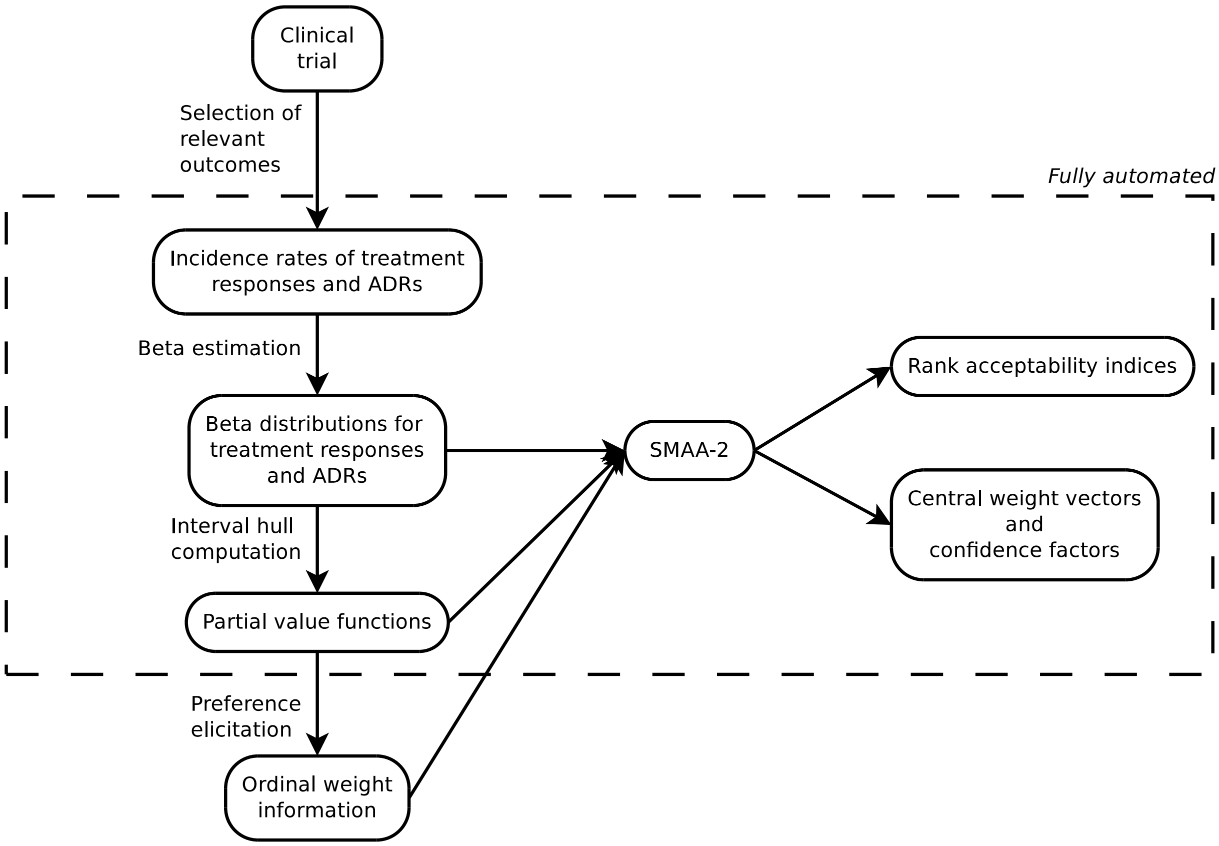

Figure 5. Model generation in application of SMAA-2 for drug benefit-risk analysis.

The following steps depend on the method. For SMAA-TRI the profiles and their measurements, criteria thresholds, and the lambda cutting level must additionally be input, and the decision is aided through the category acceptability indices. SMAA-2 can be used with or without weight information; if no weight information is used, the preferences supporting each alternative can be described with central weight vectors and their certainty assessed through confidence factors. The SMAA-2 rank acceptability indices can be used to assess imprecision of the ranking irrespective whether or not weight information is incorporated into the model. After the relevant indices have been inspected, the DM should assess whether the provided information is sufficient for taking the decision. Note that "sufficient" here is very subjective - it is up to the DM to decide whether e.g. a 80% first rank acceptability is sufficient precision to choose the alternative for implementation or if a higher acceptability is desired. If the indices are deemed sufficiently precise, the process can be terminated with decision recommendations. If not, the decision problem formulation in terms of included criteria and alternatives should be inspected. If new ones are to be added or some old ones removed, the process should be iterated in order to e.g. also add measurements for the new alternatives. In case the formulation is appropriate but the indices do not provide sufficient discrimination of the alternatives, the criteria measurements, preference- and technical parameters can be defined more exact. Note that the actual model definition does not have to be made in a linear manner, but e.g. the lambda interval can be input before the preferences.

The criteria measurements can be defined as exact real values or imprecise as intervals, ordinal ranks, or as normal (Gaussian), log-normal, logit-normal, or beta distributed. For normal and logit-normal distributed measurements also a relative version of the distribution is available, that contains an alternative-specific part and a baseline common for all alternatives on a single

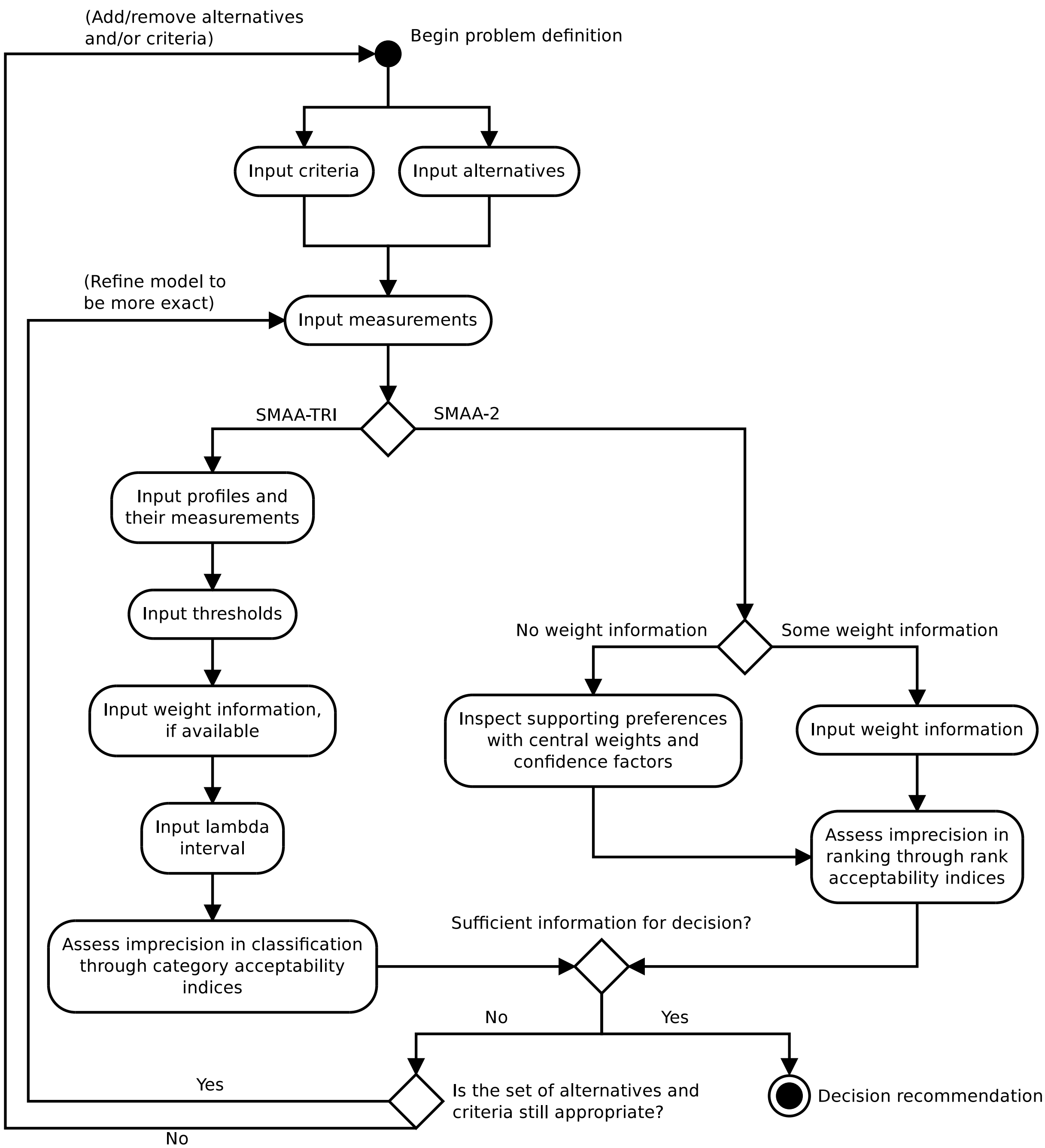

Figure 6. The standard JSMAA decision support process as a UML state diagram.

criterion (van Valkenhoef et al. 2011). JSMAA supports exact, imprecise linear constrained (interval) or ordinal preference information (weights). In case of SMAA-TRI, preference- and indifference thresholds are supported, and their values can be exact or disjunct intervals. The SMAA-TRI lambda cutting level is input as an interval. The required inputs and the provided outputs are described in Table 1.

Note that the decision process presented in Figure 6 does not include execution of the simulations. JSMAA tries to minimize the required amount of user interaction, and uses a separate thread to run the simulations in background whenever the model changes. In practice the simulation overhead is often unnoticeable to the user.

Table 1. Inputs and outputs of the methods implemented in JSMAA

Method	Inputs	Outputs
SMAA-2	Alternatives, criteria, criteria measurements, weight information	Rank acceptability indices, central weight vectors, confidence factors
SMAA-TRI	Alternatives, criteria, categories, criteria measurements (both for alternatives and profiles), thresholds, weight information, lambda interval	Category acceptability indices

4.1. *User interface*

The user interface of JSMAA is divided in two panels. On the left panel is a tree view of the model and the results. A panel on the right side shows details for the model element or results that are currently selected in the left tree. Figure 7 presents[1] the layout when the category acceptability indices of a SMAA-TRI model are selected. Note the lambda-slider in the bottom of the screen. Whereas the other parameters of the model (criteria measurements, preferences) are input in their corresponding screens, the lambda cutting level is separated to the bottom tool bar to allow easily experimenting how the results change with (imprecise) values for the technical parameter lambda. When the lambda value is changed, the whole simulation is automatically re-run (as is the case with changing other parameters of the model).

The criteria measurements can be input simultanoeusly for all criteria as shown in Figure 8. The partial value functions are displayed only for SMAA-2 models. Note that JSMAA currently supports only linear partial value functions. Preferences (weight information) can be input as either exact values, weight intervals, or ordinal (ranking) of weights. Figure 9 presents input of ordinal weight information. For SMAA-2 models, the scale swing of the partial value functions should affect the weights, so it is presented in the weight input pane.

All results are presented in tabular format and visualized with charts. SMAA-TRI category acceptability indices and SMAA-2 rank acceptabilities are presented with bar charts (see Figure 7 for the case of SMAA-TRI). SMAA-2 central weights and confidence factors are presented in the same table due to their close relation. Central weights are visualized with a line chart as shown in Figure 10. The datasets of all results' figures can be exported as GNUPlot[1] scripts for easy use in paper publications.

4.2. *Architecture*

In order to support integration with external systems, JSMAA is split in three different modules: the library containing all code for the models and their computation (jsmaa-lib), the re-usable graphical user interface components such as the results pane in Figure 7 and weight input pane in Figure 9 (jsmaa-gui), and rest of the program code (jsmaa-main). External open source libraries are used for general functionalities such as graph drawing, thread scheduling, and user interface components.

JSMAA is implemented in Java and therefore it is usable in all major operating systems (including Linux, Mac OS X, and Windows). Although it is currently the most mature SMAA implementation, it is far from complete, and extra effort has been made to allow new developers to continue the work (cf. `www.smaa.fi/jsmaa.php`). The source code is openly available in a repository at `http://github.com/tommite/jsmaa`. We hope that new developers would join

[1] All screenshots are from JSMAA v0.8.4
[1] `http://gnuplot.info`

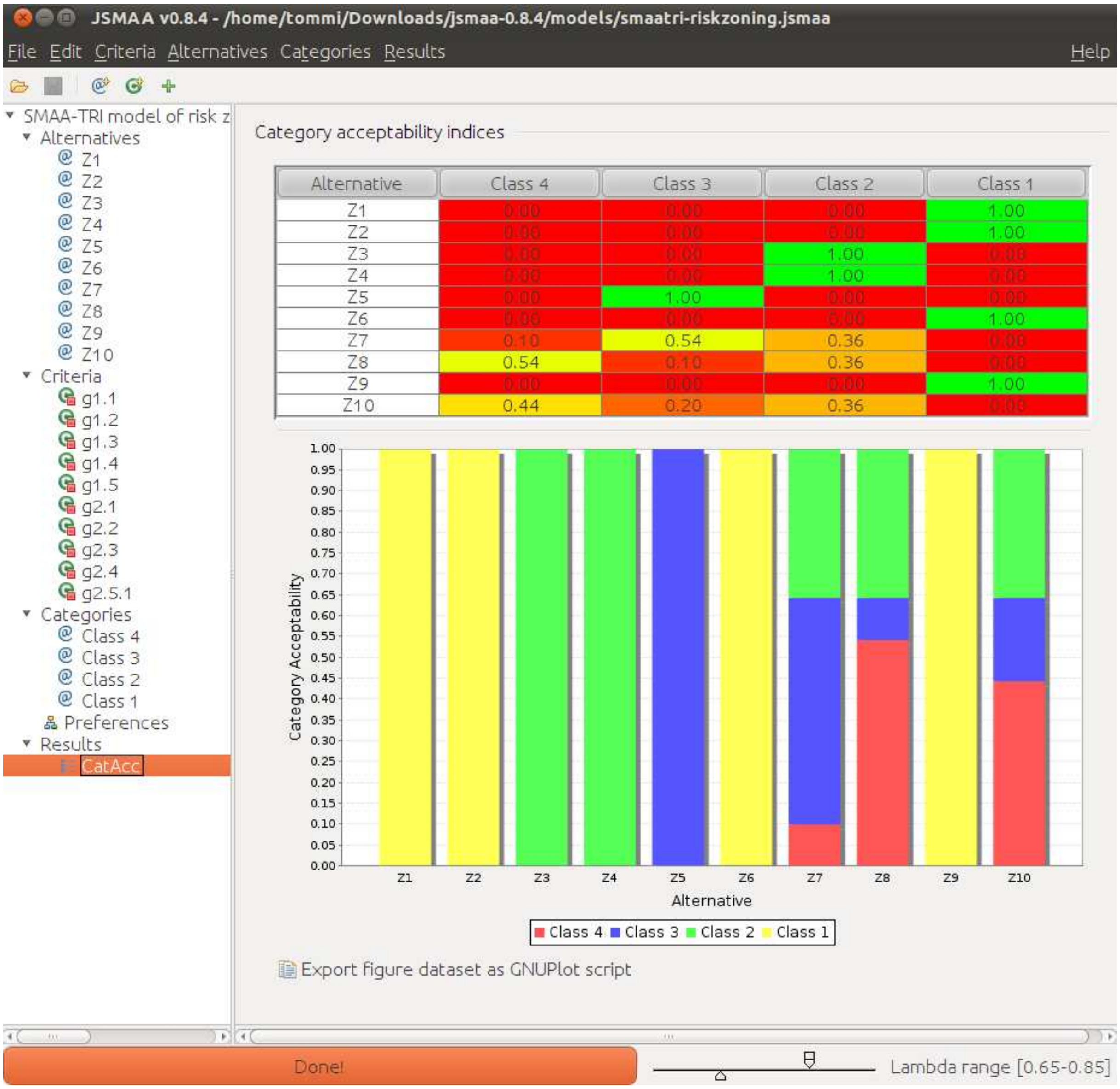

Figure 7. SMAA-TRI category acceptability indices in JSMAA.

development to add new features, especially graphical user interface components that would make the software more usable for those not familiar with MCDA.

4.3. *Model generation*

JSMAA enables model generation programatically (direct integration with jsmaa-lib and jsmaa-gui) or by writing JSMAA model files in XML. Currently a proprietary XML format is used, of which an example is given in Listing 1. For more examples, see the sample models within the distribution package of JSMAA.

Listing 1 Example JSMAA XML file format. Only one alternative, criterion, and ordinal preference statement (criterion rank) is shown.

```
<SMAA-2-model name="BR-Analysis" modelVersion="2">
  <alternatives>
    <alternative id="0" name="Venlafaxine 0.0 mg/day"/>
```

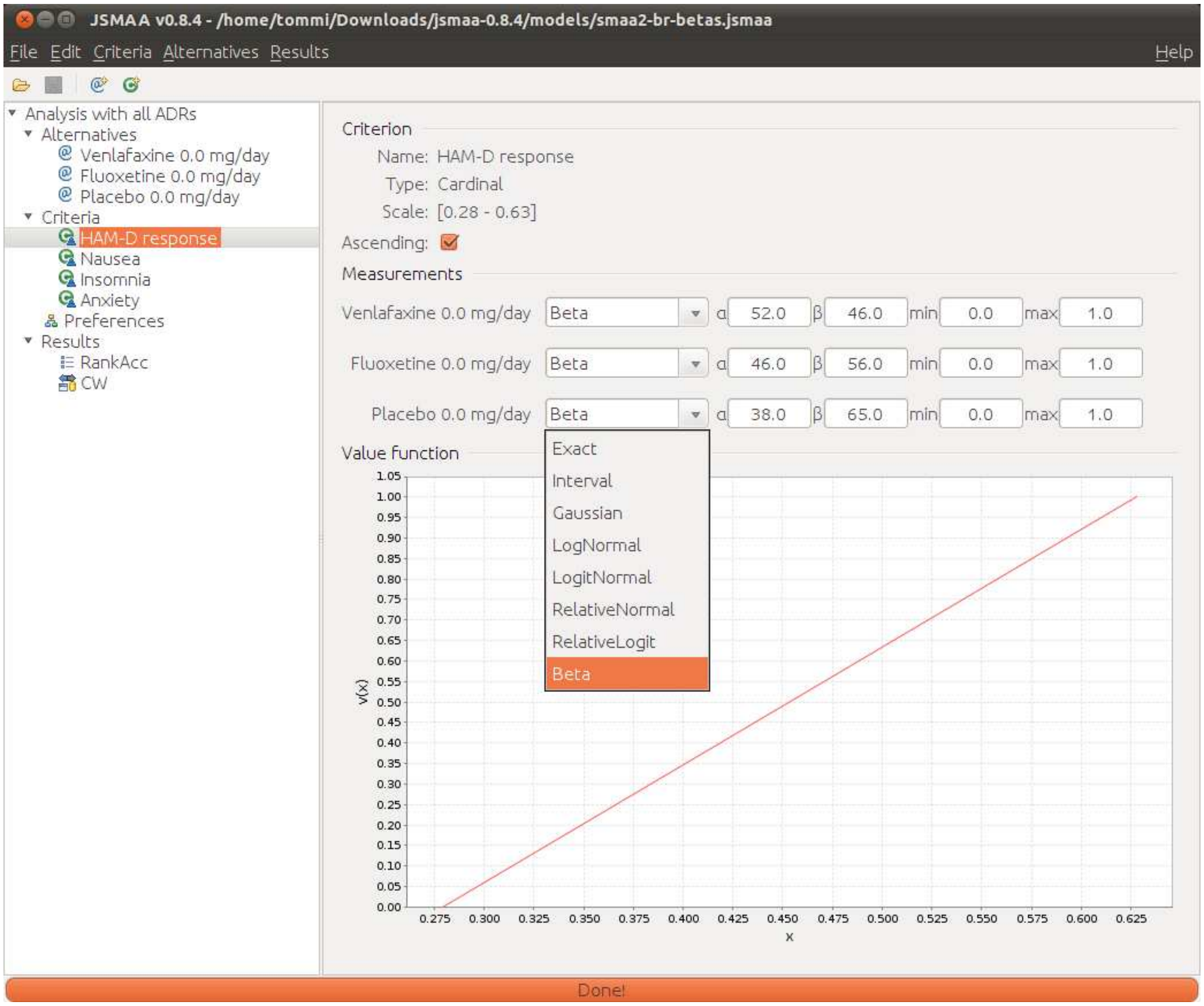

Figure 8. Input of criteria measurements in JSMAA.

```
    ...
</alternatives>
<criteria>
  <criterion class="cardinalCriterion" id="3" name="HAM-D">
      <ascending value="true"/>
  </criterion>
  ...
</criteria>
<measurements>
  <measurement>
      <criterion class="cardinalCriterion" ref="3"/>
      <value class="beta" alpha="52.0"
         beta="46.0" min="0.0" max="1.0"/>
      <alternative ref="0"/>
  </measurement>
  ...
</measurements>
<preferences class="ordinalPreferences" id="7">
```

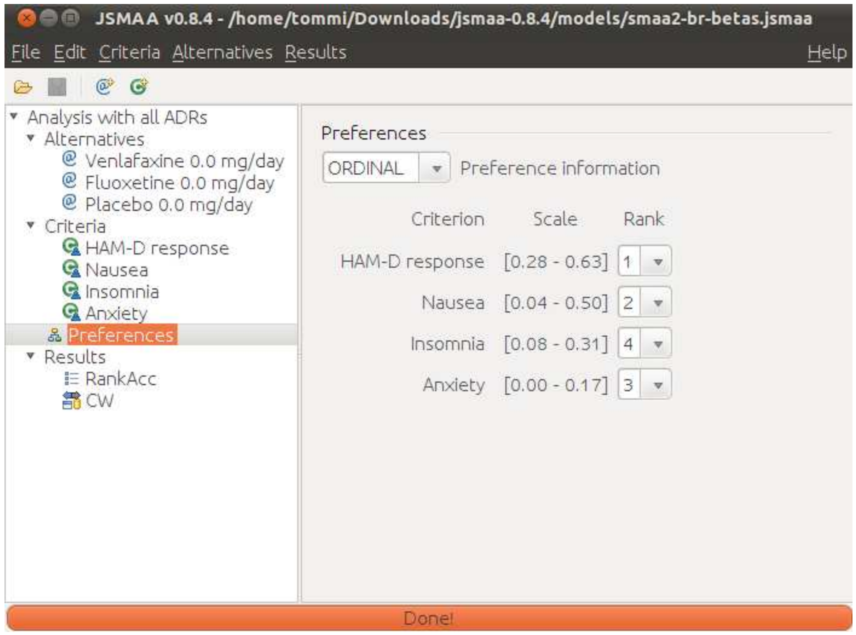

Figure 9. Input of ordinal weight information in JSMAA.

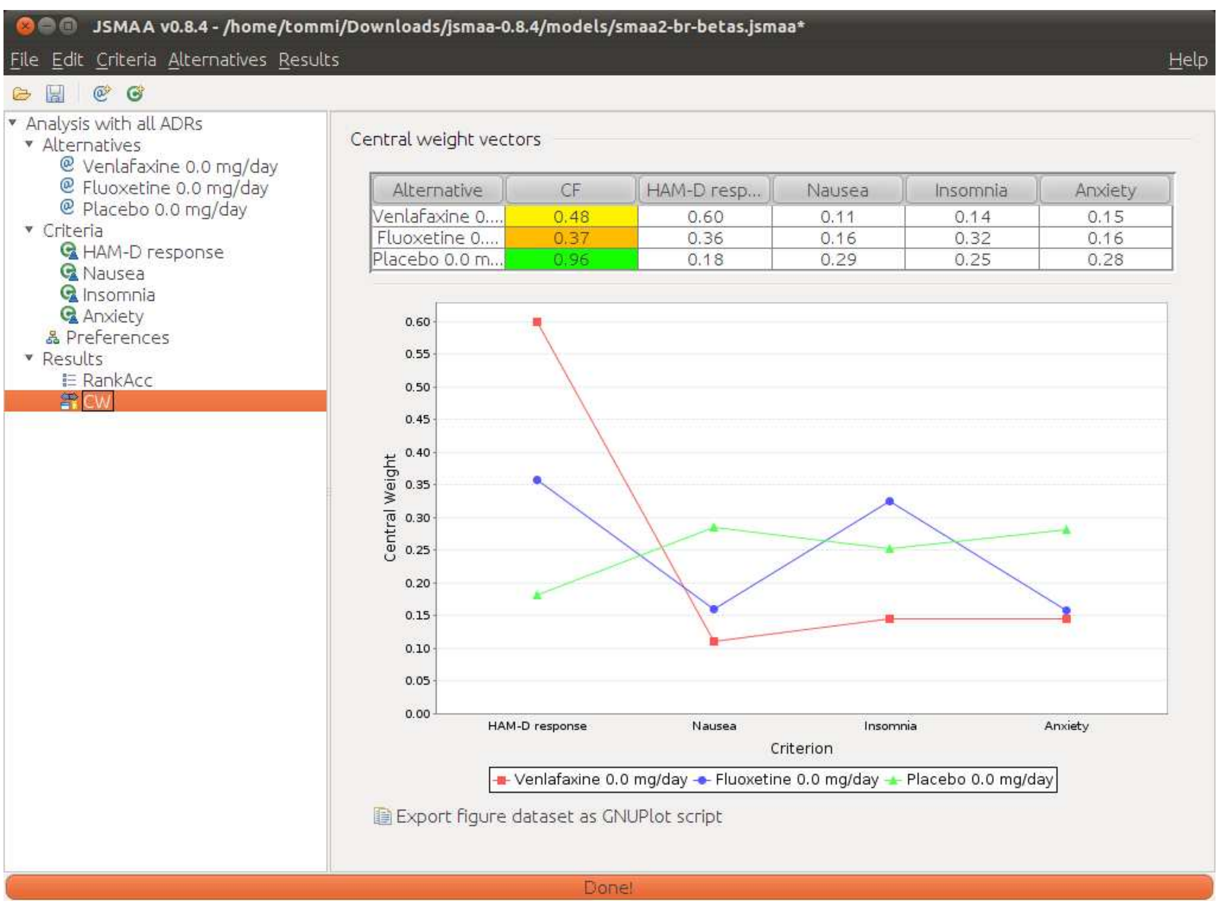

Figure 10. Visualization of SMAA-2 central weight vectors in JSMAA.

```
    <preference>
      <criterion class="cardinalCriterion" ref="3"/>
      <value class="rank" rank="1"/>
    </preference>
    ...
  </preferences>
</SMAA-2-model>
```

5. Conclusions

The lack of impact of MCDA in managerial practice is partly caused by the current implementation of MCDA methods – in proprietary software that is hard to integrate to existing information systems. Open source software and model generation can enable wider application of MCDA in new fields where the decision alternatives or criteria are generated as part of the decision support process. Most of current MCDA software do not provide an interface for integration with external systems, and therefore there exists a need for design and implementation of more modular MCDA components.

Such components can be built top-down, through an implemention of a complete method and provision of an interface for the model initialization and results visualization. This paper described a software build in such a way, JSMAA, implementing the SMAA-2, SMAA-O, and SMAA-TRI methods. Another approach is to design the components bottom-up as promoted by the Decision Deck Diviz platform (`www.diviz.org`). In Diviz, small algorithmic components communicating through an XML data standard, XMCDA, are combined to form workflows that represent MCDA methods. The bottom-up approach allows for more interactive in-depth analysis of different parts of the methods, and it seems quite suitable for educational purposes. Whether it is a proper way to design components providing interface for integration in domain-specific decision support systems is questionable. However, when XMCDA matures, it could become *the* MCDA data exchange standard enabling MCDA model sharing even across monolithic systems.

The learning experienced by the decision makers with multi-criteria analysis is considered by many experts as one of the major advantages of MCDA, and this should be taken in consideration when designing user interfaces for MCDA software tools. The model construction is dependent on the method, which has implications on how the decision makers' preference structures can and should be elicited. Care needs to be taken when MCDA components, such as the library of JSMAA, are used in domain-specific decision support systems: prerequisites of the chosen method (such as the meaning of weights, cf. Choo et al. 1999) need to be met. However, reasoning that only an MCDA expert is able to construct mathematically correct models should not be used as an excuse to refrain from developing usable and open MCDA software.

References

Belton, V., and Hodgkin, J. (1999), "Facilitators, decision makers, D.I.Y. users: Is intelligent multicriteria decision support for all feasible or desirable?," *European Journal of Operational Research*, 113, 247–260.

Belton, V., and Stewart, T.J., *Multiple Criteria Decision Analysis - An Integrated Approach*, Kluwer Academic Publishers, Dordrecht (2002).

Choo, E.U., Schoner, B., and Wedley, W.C. (1999), "Interpretation of criteria weights in multicriteria decision making," *Computers & Industrial Engineering*, 37(3), 527–541.

French, S., and Xu, D.L. (2005), "Comparison study of multi-attribute decision analytic software," *Journal of Multi-Criteria Decision Analysis*, 13, 65–80.

Keen, P.G.W., and Sol, H.G., *Decision Enhancement Services*, IOS Press, The Netherlands (2008).

Keeney, R.L., *Value-focused thinking: a path to creative decisionmaking*, Harvard University Press (1996).

Lahdelma, R., Miettinen, K., and Salminen, P. (2003), "Ordinal criteria in stochastic multicriteria acceptability analysis (SMAA)," *European Journal of Operational Research*, 147(1), 117–127.

Lahdelma, R., and Salminen, P. (2001), "SMAA-2: Stochastic multicriteria acceptability analysis for group decision making," *Operations Research*, 49(3), 444–454.

Lakhani, K.R., and von Hippel, E. (2003), "How open source software works: free user-to-user assistance," *Research Policy*, 32(6), 923–943.

Law, A., and Kelton, W.D., *Simulation Modeling and Analysis*, 3rd ed., McGraw-Hill (1999).

Malczewski, J., *GIS and Multicriteria Decision Analysis*, Wiley (1999).

Nickel, S., Puerto, J., and Rodríguez-Chía, A.M. (2005), "MCDM Location Problems," in *Multiple Criteria Decision Analysis: State of the Art Surveys* eds. J. Figueira, S. Greco and M. Ehrgott, New York: Springer Science+Business Media, Inc.

Roy, B., *Multicriteria Methodology for Decision Analysis*, Kluwer Academic Publishers, Dordrecht (1996).

Siskos, Y., and Spyridakos, A. (1999), "Intelligent multicriteria decision support: Overview and perspectives," *European Journal of Operational Research*, 113, 236–246.

Smets, P. (1991), "Varieties of ignorance and the need for well-founded theories," *Information Sciences*, 57–58, 135–144.

Spronk, J., Steuer, R.E., and Zopounidis, C. (2005), "Multicriteria Decision Aid/Analysis in Finance," in *Multiple Criteria Decision Analysis: State of the Art Surveys* eds. J. Figueira, S. Greco and M. Ehrgott, New York: Springer Science+Business Media, Inc.

Tervonen, T., and Figueira, J.R. (2008), "A survey on stochastic multicriteria acceptability analysis methods," *Journal of Multi-Criteria Decision Analysis*, 15(1–2), 1–14.

Tervonen, T., Figueira, J.R., Lahdelma, R., Almeida Dias, J., and Salminen, P. (2009), "A stochastic method for robustness analysis in sorting problems," *European Journal of Operational Research*, 192(1), 236–242.

Tervonen, T., and Lahdelma, R. (2007), "Implementing stochastic multicriteria acceptability analysis," *European Journal of Operational Research*, 178(2), 500–513.

Tervonen, T., Linkov, I., Steevens, J., Chappell, M., Figueira, J.R., and Merad, M. (2009), "Risk-based classification system of nanomaterials," *Journal of Nanoparticle Research*, 11(4), 757–766.

Tervonen, T., van Valkenhoef, G., Buskens, E., Hillege, H.L., and Postmus, D. (2011), "A stochastic multi-criteria model for evidence-based decision making in drug benefit-risk analysis," *Statistics in Medicine*, 30(12), 1419–1428.

van Valkenhoef, G., Tervonen, T., Zwinkels, T., de Brock, B., and Hillege, H.L. (2012), "ADDIS: a decision support system for evidence-based medicine," *Decision Support Systems*, to appear.

van Valkenhoef, G., Tervonen, T., Zhao, J., de Brock, B., Hillege, H.L., and Postmus, D. (2011), "Multi-criteria benefit-risk assessment using network meta-analysis," *Journal of Clinical Epidemiology*, (in press).

Wallenius, J., Dyer, J.S., Fishburn, P.C., Steuer, R.E., Zionts, S., and Deb, K. (2008), "Multiple criteria decision making, multiattribute utility theory: recent accomplishments and what lies ahead," *Management Science*, 54(7), 1336–1349.

Weistroffer, H.R., Smith, C.H., and Narula, S.C. (2005), "Multiple criteria decision support software," in *Multiple Criteria Decision Analysis: State of the Art Surveys* eds. J. Figueira, S. Greco and M. Ehrgott, New York: Springer Science+Business Media, Inc.

Yu, W. (1992), "Aide multicritère à la décision dans le cadre de la problématique du tri: Concepts, méthodes et applications," Université Paris-Dauphine.