Application of Multi-Criteria Decision Analysis to the Selection of Software Measures

Yoan Pacheco Cardenas

Universidad Central "Marta Abreu" de las Villas, Santa Clara,
Cuba

yoan@uclv.edu.cu

Abstract. In this research we propose the application of a multi-criteria decision analysis to make documented and transparent decisions about software measures' selection. The Pareto's dominance method was utilized to narrow down the initial measures' list. The multi-attribute value theory was applied for ranking the final set of measures. As a result there was eliminated about 40% of the initial measures and the final measures' list was ranked.

Keywords. Multi-criteria decision analysis, multi-attribute value theory, selection of software measures.

1 Introduction

In the context of software product evaluation, quality characteristics and sub-characteristics are evaluated by a quality measure [1]. Thoughtful, appropriate measures selection is an important step for effectively evaluating a software product among a list of alternatives. Clearly defined software measures increase knowledge of the software product and asses its usefulness by creating a targeted, effective means of evaluation. The variety and complexity of software products produces a multitude of potential measures. For example, Graham provides 80 different measures for evaluating a business rules management system (BRMS) [2]. However, with limited funding, it may not be possible to effectively evaluate all measures, so it is critical to select the set of measures that can most clearly indicate the potential of the software product in relation to evaluation goals. Measures' selection is thus a complex process. There are a number of commonly used methods for measures' selection, including previous case studies [3], judgments of stakeholders and experts, screening using established criteria sets, conceptual modeling and Analytic Hierarchy Process (AHP) [4]. However, the increase of the projects' complexity, the necessity of transparence on decisions and the need of an effective process for elicitation of stakeholder's opinion suggest that the efficacy of former measures' selection methods can be improved.

In this work we propose the application of multi-criteria decision analysis (MCDA), to quantitatively evaluate software measures based on their value for stakeholders with respect to defined criteria, and the relative importance of those criteria [5].

The MCDA methods have been extensively applied to select ecological indicators in environmental case studies [6, 7, 8]. Nevertheless, to the best of our knowledge, an MCDA approach was never used to select software measures. We believe that a formal MCDA-based method can be very useful for the selection of measures that can be used in evaluating software products. It will enable software product evaluators to make methodological and transparent decisions.

The paper is structured as follows. Previous work describes the most commonly used methods, and its limitations for measures' selection. Materials and Methods describe the case study and the development of the components of the MCDA model. Results and Discussion present the results of the domination analysis and measures' list ranking. In the Conclusions and future work section are discussed the benefits and limitations of utilizing MCDA for software product measures' selection and the most appropriate circumstances in which to apply this methodology. This section also describes the futures steps of this research.

ISSN 1405-5546
doi: 10.13053/CyS-22-1-2765

2 Previous Work

Software measures' selection methods include the use of previous case studies, judgments of stakeholders and experts, screening, conceptual modeling and AHP. Previous case studies are based on the selection of measures used previously for measuring similar software products. However, for an effective use of this method, the organization must have a reliable and comparable set of measures of its projects [3]. When no previous case studies are available to undertake quantitative evaluations, the judgments of stakeholders and experts can be useful in carrying out short-listing of measures. Initially, a research on software quality frameworks and assessments of the software program is conducted to identify an initial measures' list.

The measures' list then is refined. This should be done in consultation with stakeholders and experts. However, measures' selection via this method may exclude or place bias on specific stakeholders or experts values. For small projects this method is generally inexpensive and time-efficient. In complex projects hundreds of measures may be identified. This process can be very time consuming and exceedingly difficult.

Another problem of this method is the lack of transparency, which makes the decision-making process more difficult to justify and document. As a more transparent alternative or supplement to previous case studies and best professional judgment, software project managers may sometimes evaluate or ”screen” potential measures against a set of criteria to identify the most appropriate subset of measures for a given software program.

Screening is relatively inexpensive and time-efficient but is generally not adequate as a standalone method. Screening does not have a quantitative internal structure for determining whether a measures' set is comprehensive. This method is based on judging measures against some “evaluation criteria” which are identified subjectively. Such evaluations are therefore likely to be biased and context dependent. The conceptual model method can provide stakeholders with a clear view of important factors and their relationships, making it easier to develop a measure set.

These relationships reveal which attributes' project measures should aim to assess. A conceptual or domain model is a visual representation of conceptual classes or real-world objects in a domain of interest [23, 24]. It may show domain objects or conceptual classes, associations between conceptual classes and attributes of conceptual classes [25].

The conceptual modeling approach for selecting measures does not assign weights or prioritize the model components. Therefore, this approach does not help in making trade-offs between measures of the same or different components.

The conceptual modeling method still leaves room for bias, as stakeholders often participate in the development of the model. Conceptual models are simplifications that usually focus only on the components considered most relevant while leaving out other components less important or less understood. AHP has been used in several software selection problems [26]. It is based on a subjective pairwise comparison of criteria and it has been criticized for its rank reversal, measurement scale, and transitivity of preferences [7]. Compared to the common measures' selection methods presented before (previous case studies, best professional judgment, screening using established criteria sets, conceptual modeling, and AHP), MCDA is more comprehensive and inclusive, incorporating stakeholder preferences from several subjects and fields.

This method allows software evaluators to simplify complex situations with several objectives and alternatives under consideration. Stakeholders can review components of the model including weights and measures' scores, and decision makers can justify management choices according to model results. The MCDA method for measures' selection thus enables software product evaluators to make methodological and transparent decisions. The quantitative results allow decision makers to easily compare each alternative and to select the optimal measures' set. MCDA can be extremely useful, but it also has some limitations. It can be time consuming and more costly than other simpler measures' selection methods. It takes a substantial amount of work and expert judgment to assign value scores to each alternative for every criterion.

Small increments in the quantity of evaluation criteria and alternatives result in much larger increases in necessary input information. For example, in this case study 31 measures were evaluated with respect to seven criteria and five sub-criteria. This required 372 expert evaluations of the value of each alternative for every criterion.

3 Materials and Methods

3.1 Case Study Details

In partnership with a local software company of Santa Clara City, the Database Group of the Universidad Central “Marta Abreu” de Las Villas, Cuba planed the evaluation of a group of BRMS with the goal of selecting the best program in terms of quality and cost.

A multi-disciplinary stakeholders group (Information technology users, business users, BRMS technical consultant and managers), was assembled to set objectives, formulate and evaluate alternatives for BRMS program selection. The evaluation task depends on selecting the most appropriate measures to assess how well the project’s objectives are accomplished by software products. The stakeholders group chose to use the multi-criteria decision analysis (MCDA), methodology to guide their selection of the optimal measure set as the project involves a complex system with multiple objectives and stakeholders.

3.2 Stages of the Multi Criteria Decision Analysis Process

In the context of MCDA the software quality characteristics are part of the evaluation criteria and the measures are the alternatives to select and rank. We divided the MCDA process in six stages:

1. Definition of the set of possible measures: Stakeholders should create the potential measures’ pool. In this case study, initial selection of measures was done after discussions with the stakeholders, reviewing the literature for BRMS software evaluation [2,3,9-14], studying project objectives and the ISO/IEC 912615 and SQuaRE series of software product quality standards [1,16]. The potential measures’ set was organized by criteria and sub-criteria, see Table 1.
2. Definition of the evaluation criteria and sub-criteria: Stakeholders should identify evaluation criteria. This step was done after analyzing project objectives, the components and users of a BRMS and the ISO/IEC 9126 and SQuaRE series of software product quality standards. There were selected criteria that evaluate three groups of software characteristics, specifically: managerial characteristics, product quality and quality in use. Cost criteria were selected for the evaluation of managerial characteristics. For the evaluation of BRMS product quality characteristics, stakeholders selected functional suitability, reliability, performance efficiency, security and maintainability criteria. Functional suitability has three sub-criteria that describe more specifically the components of a BRMS, the rules engine, the rules repository and a group of management tools. Usability criteria were selected for the evaluation of quality in use characteristics. It is divided in two sub-criteria that describe more specifically the types of users of a BRMS: business users and information technology (IT) users, see Table 1.
3. Definition of the value of each alternative measure with respect to each criterion: In the context of this project, we used a value function for each criterion. This value function spans from 0 to 1, with a value of 1 being assigned to the value of the best alternative score for that criterion and 0 being assigned to the value of the worst alternative score. In this case, a linear value function was used, which assumes that increases in value are directly related to increases in the alternative’s score for the criterion in question. Finally, there was assigned to each measure a value score based on stakeholders assessment of the measure’s ability to provide useful information about each of the criteria and sub-criteria.
4. Running MCDA ”domination analysis”: A feasible combination of measures for a collection of objectives is said to be Pareto dominated if there does exist another feasible combination of measures under which each objective is, at least, as well off and some objective is strictly better off [17]. We utilized the MCDA software Decerns (Decision Evaluation

in ComplEx Risk Network Systems), to model the problem space and for the domination analysis [18]. Specifically, Decerns implements a Pareto dominance method. Pareto-based domination analysis was used to reduce the number of the potential measures based only on their value with respect to each criterion.

5. Definition of the weight of each criteria and sub-criteria: There was used a Max100 direct rating approach for weighting elicitation. It is a reliable and relatively simple to use method. It is also preferred by interviewed persons [19]. Four stakeholder subgroups were considered: information technology users, business users, BRMS technical consultant and managers. We selected three persons from each of the four stakeholder groups and directed an interview with each person for approximately 1.0 h. The interviewer gave the person a copy of the set of the initial BRMS criteria and sub-criteria, see Table 1. The interviewee organized the criteria by importance level and next indicated the relative importance of the criteria by rating them along a 100 point scale. Starting with the interviewee most important criterion, the interviewee positioned each of the criteria along the 100 point scale. These procedures are repeated for each level of sub-criteria until all levels were completed. For each group we calculate the average criteria weights. Finally, the weight normalization was done.
6. Running the MCDA MAVT method to rank the measures' list: Decerns software was also used for running the MCDA MultiAtribute Value Theory (MAVT), method and for weight sensitivity analysis [5, 20-22]. The MCDA MAVT approach was used to rank the measures' list in terms of their overall value function. Weight sensitivity analysis was utilized for understanding the influence of business user's usability weight on the output results. In particular, for distributing the measures of business users' usability in groups.

3.3 MCDA MAVT Method

Techniques, that under certainty use a value function that spans from 0 to 1, to compose a persons' preference of an attribute into a value are part of the MAVT methods. These methods aggregates the preferences of the evaluator into a function F() to form an overall evaluation. The most simple and used form of F() is the additive form. In this approach is calculated a weighted summation of the performance of each alternative against all the criteria. The objective of the decision maker is to select the alternative that maximizes the value of F(). This is the procedure used by Decerns. For a correct implementation of the additive model in MAVT the decision maker must be rational, prefer more value to less value and be consistent in his judgments. In MAVT a bad performance in some criteria can be compensated by a good performance in other criteria.

4 Results and Discussion

4.1 Domination Analysis

Dominated measures were those that were outperformed (had lower value scores), by at least one other measure in all criteria. These measures were eliminated as they would not be selected regardless of the assigned weights. The result of this step was a smaller set of non-dominated measures which were then analyzed and ranked. The initial Pareto-based domination analysis eliminated 40.3% of the potential measures based only on their value with respect to each criterion. Specifically, this resulted in the identification and elimination of 21 dominated measures, narrowing the measures' pool from the initial set of 52 software product measures to the 31 non-dominated measures, see Table 2.

This greatly simplifies the decision and provides a clear justification for removing dominated measures independently of stakeholder preferences as they are sub-optimal under any set of weights. The Pareto-dominance method has a mathematical basis that converges to efficient solutions, but also may lead to inequitable results. Special care should be observed when social or environmental measures are part of the pool of potential measures. Measures of software quality or cost may dominate to social or environmental measures. It may produce an efficient but not ethic or ecological solution. Therefore, the results of the domination analysis should be carefully analyzed.

Table 1. The criteria, sub-criteria and potential measures

Criteria	Sub-criteria	Potential measures	
Cost		1-Product license 2-Training	3-Maintenance 4-Support services
Functional suitability	Rules engine	5-Backward and mixed chaining 6-RETE algorithm 7-Multiplatform	8-XML input 9-Interfaces with C,C++,Java and .Net 10-Runtime rule updates
	Rules repository	11-Change management 12-Version control features 13-Web interface	14-Ability to organize rule groups/sets 15-User friendly repository interface 16-Hot deployment
	Management tools	17-Decision tables 8-Decision trees 19-Rules in natural language	20-Ability to specify test cases 21-Ability to execute test cases 22-Available plugins
Reliability		23-Maturity in rule engine market	24-Fault tolerance 25-Recoverability
Usability	IT Users	26-Need to leverage technical skills 27-Java/.Net integration 28-Coding of rules in Java/.Net 29-Active developers community	30-Documentation for developers 31-Code examples 32-Web services 33-Debugging of rules
	Business users	34-Report generation capabilities 35-Learning curve 36-Multilanguage Support	37-Organizational vocabulary 38-User manuals 39-Tutorials 40-User interface
Performance efficiency		41-Time behavior 42-Memory consumption	43-Handling of large number of rules
Security		44-Confidentiality 45-Integrity 46-Authentication	47-Accountability 48-LDAP integration
Maintainability		49-Modularity 50-Reusability	51-Analyzability 52-Open Source

Table 2. Pareto-based domination analysis

Measures name	Dominated by
3- Maintenance	52- Open Source
4- Support services	52- Open Source
50- Reusability	52- Open Source
51- Analyzability	52- Open Source
8- XML input	9- Interfaces with C, C++, Java and .Net
27- Java/.Net integration	9- Interfaces with C, C++, Java and .Net
14- Ability to organize rule groups/sets	15- User friendly repository interface
20- Ability to specify test cases	21- Ability to execute test cases
24- Fault tolerance	25- Recoverability
11- Change management	12- Version control features
13- Web interface	40- User interface
30- Documentation for developers	38- User manuals
31- Code examples	39- Tutorials
32- Web services	27- Java/.Net integration
37- Organizational vocabulary	34- Report generation capabilities
28- Coding of rules in Java/.Net	19- Rules in natural language
26- Need to leverage technical skills	35- Learning curve
42- Memory consumption	43- Handling of large Number of rules
44- Confidentiality	46- Authentication
45- Integrity	46- Authentication
47- Accountability	46- Authentication

The Figure 1, represents an MCDA decision tree showing overall project objective, criteria, sub-criteria, and measures' list for the case study and how they are related and structured within the MCDA framework.

Here the overall goal is to rank measures. Seven main project criteria or objective categories are presented, including cost, functional suitability, reliability, usability, performance efficiency, security and maintainability.

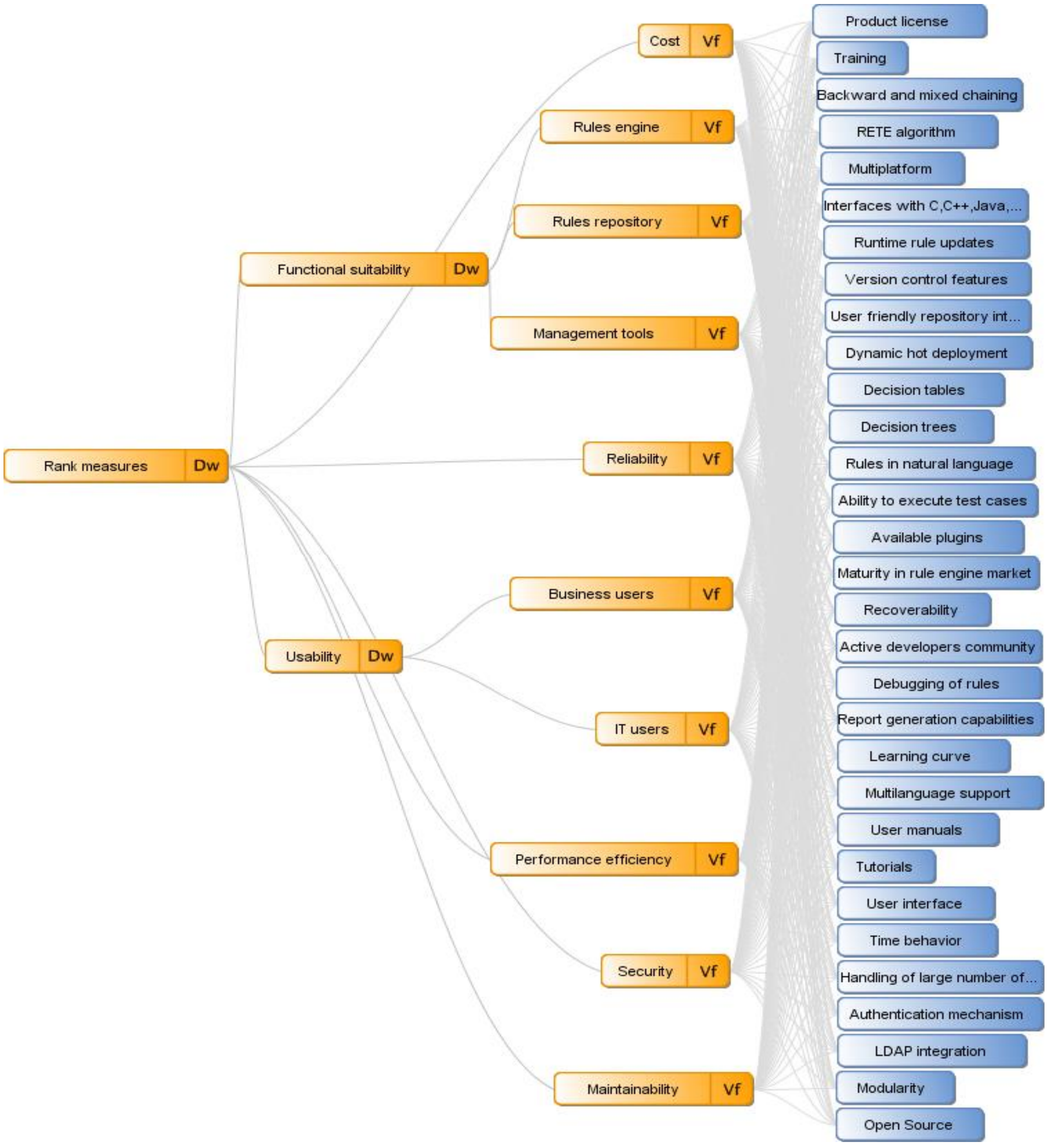

Fig. 1. Value tree of project objective, criteria, sub-criteria and non-dominated measures

The main criteria then have sub-criteria that describe more specific categories of objectives (e.g. functional suitability is split into rules engine, rules repository and management tools), so that each of these objective categories may be weighted and scored separately. The right side of the model shows the measures' choices that the model will evaluate with respect to the criteria. The value, and therefore the ranking of each measure, is a function of the ability of each measure to describe the criteria and sub-criteria (value score) and the relative importance of describing those criteria and sub-criteria (weights).

4.2 Definition of the Weight of Each Criteria and Sub-Criteria

Stakeholders were interested in reducing project costs and therefore assigned the higher weight to cost criterion.

Table 3. Measures ranked by MAVT Method

Rank	Measure
1	1-Product license
2	2-Training
3	6-RETE algorithm
4	5-Backward and mixed chaining
5	9-Interfaces with C, C++,Java and .Net
6	12-Version control features
7	15-User friendly repository interface
8	17-Decision tables
9	19-Rules in natural language
10	10-Runtime rule updates
11	7-Multiplatform
12	22-Available plugins
13	21-Ability to execute test cases
14	18-Decision trees
15	16-Hot deployment
16	49-Modularity
17	46-Authentication
18	52-Open Source
19	3-Handling of large number of rules
20	41-Time behavior
21	48-LDAP integration
22	33-Debugging of rules
23	29-Active developers community
24	40-User interface
25	38-User manuals
26	34-Report generation capabilities
27	39-Tutorials
28	35-Learning curve
29	36-Multilanguage support
30	25-Recoverability
31	23-Maturity in rule engine market

The normalized weights assigned by stakeholders are: 0.24 for cost, 0.20 for functional suitability, 0.16 for performance efficiency, 0.15 for maintainability, 0.10 for security, 0.08 for reliability and 0.07 for usability.

For functional suitability sub-criteria: 0.40 for rules engine, 0.30 for rules repository and 0.30 for management tools. For usability sub-criteria the weights are: 0.60 for IT users and 0.40 for business users.

These normalized weights always sum to one in criteria and sub-criteria. Weights are highly dependent on which stakeholders' views are incorporated, so it is critical to involve a variety of stakeholders to capture all of preferences for the project's outcomes. In general, the aggregated weights representing the stakeholders of the project are the average of all stakeholder weights assigned to each criterion [5].

4.3 Running the MCDA MAVT Method to Rank the Measures' List

The results of running the MCDA analysis to rank the measures' list, that is, the average value score for each measure is represented in the same range [0, 1], see Table 3. The measures of rules engine functional suitability criteria are grayed. This type of visualization allows analysts to easily compare the value of each measure as calculated by MAVT method. As many of the scores are similar to each other, the objective of this ranking is not to explicitly determine which measures to use but it is an excellent guide for evaluators and clearly indicates that some measures are more suitable than others.

This analysis can be done with all measures and with measures that are inside each criteria or sub-criteria. For example, among all measures, product license is clearly more useful than maturity in rule engine market. In rule engine functional suitability sub-criteria, a group of two measures (multiplatform and runtime rule updates), are ranked at positions 10 and 11, and are five positions below of the rest of measures for this sub-criteria (RETE algorithm, backward and mixed chaining and interfaces with C, C++, Java, .Net).

This indicates that the first group of measures is more suitable for measuring the rule engine functional suitability.

Once the measures' ranking is formulated the decision about how many measures to use should be a function of the available resources for evaluating those measures.

4.4 Sensitivity Analysis

The MCDA methods also allow to systematically modify a variable for determining its impact on the outcome. This technique is known as sensitivity analysis. In this case study, we increased the weighting of usability for business users' sub-criteria to demonstrate the use of the sensitivity analysis for distributing the measures of sub-criteria in groups. With an initial weight of 0.40 assigned by stakeholders, measures for business users' usability are ranked one after other from position 24 to 29 and below all IT users' usability measures when all criteria are considered.

As the weighting placed on business users' usability (and thus its value in the outcome) is increased by just over 0.5, four usability measures improved their rank in two positions (user interface, user manuals, report generation capabilities and tutorials) and two remain at the same rank, 28 and 29 (learning curve and multi-language support). Clearly, the group of measures that improved its rank is more suitable for evaluating business users' usability than the group that remains at its initial rank, see Table 4.

On the contrary, IT users' usability measures are deteriorated in four positions and descend to ranks 26 and 27, in the middle of the two groups of business users' usability measures. This result suggests that, when business users' usability is more important than information technology users' usability, the sub-criteria business users' usability can be divided in two ordered groups of measures. Therefore, sensitivity analysis can be used by stakeholders for prioritizing the evaluation of groups of measures in case of lack of resources. This result is useful only if changes in weight do not compromise project objectives.

Table 4. Ranking of measures before and after increases in business users' usability weight

Rank	weight = 0.40	weight > 0.50
22	33-Debugging of rules	40-User interface
23	29-Active developers community	38-User manuals
24	40-User interface	34-Report generation capabilities
25	38-User manuals	39-Tutorials
26	34-Report generation capabilities	33-Debugging of rules
27	39-Tutorials	29-Active developers community
28	35-Learning curve	35-Learning curve
29	36-Multilanguage support	36-Multilanguage support

5 Conclusions and Future Work

In this work we proposed the application of a six stages MCDA process to quantitatively evaluate software measures based on their value for stakeholders with respect to defined criteria, and the relative importance of those criteria. This approach, to the best of our knowledge, was never used to select software measures.

As the result, there was eliminated about the 40% of the initial measures and the final measures' list was ranked and grouped. The stakeholders defined the set of possible measures alternatives, the evaluation criteria and sub-criteria and the value of each measure with respect to each criterion.

It was ran a Pareto's domination analysis to narrow down the initial measures list. There was used a relatively straightforward to apply approach for weighting elicitation. There was increased the weighting of usability for business users sub-criteria to demonstrate the use of the sensitivity analysis for distributing the measures of a sub-criteria in groups.

The proposed MCDA-based method also has some limitations, it can be very time consuming and that is why a reasonable amount of time should be available for running it. Successful method application can be compromised by the availability of stakeholders and experts.

As strengths, the proposed MCDA-based method may serve as guide for software evaluators to make methodological, documented and transparent decisions about software measures' selection. It also allows prioritizing software measures in case of lack of resources.

As future work the Database Group is planning the development of a methodology to select the appropriate MCDA method for software evaluation. In this research we used the MAVT method but in other context a different MCDA method can be suitable.

Acknowledgements

The authors acknowledges the support of the Decerns team (lead by Professor Boris Yatsalo who also made valuable advices and comments about the paper) for allowing the use of the Decerns software. The authors also thanks the Lic. Greta Milena Mazorra Cubas for the help provided in correcting the English language of the paper.

References

1. **ISO (2005).** *ISO/IEC 25000 Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) Guide to SQuaRE*. International Organization for Standardization.
2. **Graham, I. (2007).** *Business Rules Management and Service Oriented Architecture*. Wiley.
3. **Kitchenham, B., Linkman, S., & Law, D. (1997).** Desmet: a methodology for evaluating software engineering methods and tools. *Computing & Control Engineering Journal*, Vol. 8, No. 3, pp. 120–126.
4. **Saaty, T.L. (1980).** *The analytic hierarchy process: planning, priority setting, resource allocation*. McGraw-Hill International Book Co.
5. **Keeney, R.L. & Raiffa, H. (1976).** *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons.
6. **Linkov, I. & Moberg, E. (2011).** *Multi-Criteria Decision Analysis: Environmental Applications and Case Studies*. CRC Press.
7. **Yatsalo, B., Kiker, G.A., Kim, J., Bridges, T.S., Seager, T.P., Gardner, K., Satterstrom, F.K., & Linkov, I. (2007).** Application of multicriteria decision analysis tools to two contaminated sediment case studies. *Integrated Environmental Assessment and Management*, Vol. 3, No. 2, pp. 223–233. DOI: 10.1897/IEAM_2006-036.1.
8. **Convertino, M., Baker, K., Vogel, J., Lu, C., Suedel, B., & Linkov, I. (2013).** Multi-criteria decision analysis to select metrics for design and monitoring of sustainable ecosystem restoration. *Ecological Indicators*, Vol. 26, pp.76–86. DOI: 10.1016/j.ecolind.2012.10.005.
9. **Azevedo, L. & Duarte, D. (2010).** Evaluating tools for execution and management of authorization business rules. *Revista Eletrônica de Sistemas de Informação,* Vol. 9, No. 2, pp. 1–22.
10. **Duda, J. & Stawowy, A. (2010).** A possibility of business rules application in production planning. *Archives of Foundry Engineering*, Vol. 10, No. 2, pp. 27–32.
11. **Mouromtsev, D. & Kolchin, M. (2012).** Using drools rule-platform for the optical cad web-application development. *11th Conference of Fruct Association*, pp.100–106.
12. **Pacheco-Cárdenas, Y., Rodríguez-Rodríguez, E., & González-González, L. (2014).** Evaluation of open source systems for managing business rules in enterprise scenarios. *Rev. Téc. Ing. Univ. Zulia,* Vol. 37, No. 3, pp. 262–269.
13. **Rymer, J. & Gualtieri, M. (2008).** *The Forrester wave: Business rules platforms.* Forrester Research Inc.
14. **Rymer, J., Gualtieri, M. (2011)**. *Market overview: Business rules platforms.* Forrester Research Inc.
15. **ISO. (1991).** *ISO/IEC 9126 Information Technology -Software product evaluation- quality characteristics and guidelines for their use.* International Organization for Standardization.
16. **ISO. (2007).** *ISO/IEC 25001 Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) Planning and Management.* International Organization for Standardization.
17. **Emmerich, M. & Deutz, A. (2006).** *Multicriteria Optimization and Decision Making, Principles, Algorithms and Case Studies.* Leiden Institute of Advanced Computer Studies.
18. **Yatsalo, B., Didenko, V., Gritsuk, S., & Sullivan, T. (2015).** Decerns: a framework for multicriteria decision analysis. *International Journal of Computational Intelligence Systems*, Vol. 8, No. 3, pp. 467–489. DOI: 10.1082/18756891.2015. 1023586.
19. **Bottomley, P.A. & Doyle, J.R. (2001).** A comparison of three weight elicitation methods: good, better, and best. *Omega*, Vol. 29, No. 6, pp. 553–560. DOI: 10.1016/S0305-0483(01)00044-5.

20. **von Winterfeldt, D. & Edwards, W. (1996).** *Decision Analysis and Behavioral Research.* Cambridge University Pres.
21. **Belton, V. & Stewart, T.J. (2002).** *Multiple criteria decision analysis: An integrated approach.* Kluwer Academic Publishers.
22. **Figueira, J., Greco, S., & Ehrgott, M. (2005).** *Multiple Criteria Decision Analysis: State of the Art Surveys.* Springer Science+Business Media.
23. **Martin, J. & Odell, J. (1995).** *Object-Oriented Methods: A Foundation.* Prentice Hall.
24. **Fowler, M. (1996)**. *Analysis Patterns: Reusable Object Models.* Addison-Wesley.
25. **Larman, C. (2004).** *Applying UML and Patterns.* Addison Wesley Professional.
26. **Jadhav, A.S. & Sonar, R.M. (2009).** Evaluating and selecting software packages: A review. *Information and Software Technology*, Vol. 51, No. 3, pp. 555–563. DOI: 10.1016/j.infsof.2008.09.003.

Article received on 13/08/2016; accepted on 09/10/2016.
Corresponding author Yoan Pacheco Cardenas.