

# Rote: A Tool to Support Users in Defining the Relative Importance of Quality Characteristics

Agnes Lisowska and Nancy L. Underwood

ISSCO/TIM/ETI

University of Geneva, Boulevard du Pont-d'Arve,  
CH-1211 Geneva 4,  
Switzerland

E-mail: Agnes.Lisowska@issco.unige.ch, Nancy.Underwood@issco.unige.ch

## Abstract

This paper describes the Relative Ordering Tool for Evaluation (ROTE) which is designed to support the process of building a parameterised quality model for evaluation. It is a very simple tool which enables users to specify the relative importance of quality characteristics (and associated metrics) to reflect the users' particular requirements. The tool allows users to order any number of quality characteristics by comparing them in a pair-wise fashion. The tool was developed in the context of a collaborative project developing a text mining system. A full scale evaluation of the text mining system was designed and executed for three different users and the ROTE tool was successfully applied by those users during that process. The tool will be made available for general use by the evaluation community.

## 1. Background

The approach to software evaluation expounded by ISO (ISO/IEC, 2001) and followed by the current authors centres around the building of a quality model. Such a model is a hierarchy (a tree) of software quality characteristics divided into sub-characteristics (sub-trees) which eventually bottom out into metrics which can be applied directly to the software in question. These characteristics and their associated metrics should be designed to reflect the stated or implied needs of the user of the software.

Building such a quality model and applying the relevant metrics however is only part of the evaluation story. The raw results of applying individual evaluation metrics need to be interpreted and combined to produce an overall evaluation result. In some cases however, multiple users of the same system can have very different needs. Consequently, not all characteristics are equally important for all users and so not all metrics contribute equally to the evaluation of a component or system for some of users. Therefore assessment criteria, which determine how to summarise and weight the results of applying individual metrics need to be elicited from users.

As well as indicating whether they consider a particular system attribute to be e.g. mandatory or optional for their needs, users should also indicate the relative importance of characteristics which are siblings in the quality model hierarchy. This applies at all levels of the quality model from simple low-level attributes to more complex characteristics. For example, a user may consider both a *save* and a *save as* option as mandatory for their requirements but nevertheless consider the *save as* option to have a higher relative importance since it permits saving files to different locations and as different file types. Similarly, although a user considers both an ontology management system and the facility to build and maintain ontologies to be mandatory, since he already has a very comprehensive legacy ontology, the performance of tools for acquiring new concepts is less important than the quality of the management system. See Underwood & Lisowska (2006) for a fuller

description of building a specific quality model and eliciting assessment criteria from users.

The tool described in this paper was developed during work on the Parmenides<sup>1</sup> project whose aim was to develop and evaluate a complex temporal text-mining system (Spiliopoulou *et al* 2004). The purpose of the tool was to support users in assigning relative importance to different characteristics in the quality model by the simple means of producing ordered lists.

## 2. The Relative Ordering Tool (ROTE)

In order to use ROTE for ordering quality characteristics and metrics the system must first be populated with the contents of the quality model. This is currently achieved using two files defining the structure and contents of the model. The first file contains the characteristics and metrics relevant to the evaluation, structured according to the quality model tree which has been defined for the application (the example of the Parmenides system is shown in Figure 1 below). The second file contains the descriptions of the characteristics and/or metrics being compared (those found in the first file). These are necessary in order to help the user decide which characteristics are more important to him, as simple titles of characteristics often do not provide enough information to make an informed decision.

### 2.1. General Functionality of the Tool

The general functionality of ROTE can best be described with reference to what can be seen in the user interface. Figure 1, above, shows a part of the ROTE interface containing the quality model of the Parmenides text mining system. There are five main "areas" in the ROTE interface, indicated by numbers on the figure.

The large window near the top (area 1) provides an overview of the whole quality model and allows the user to browse the tree to see the various characteristics at the different levels and to select which characteristics to sort.

---

<sup>1</sup> <http://www.crim.co.umist.ac.uk/parmenides/>

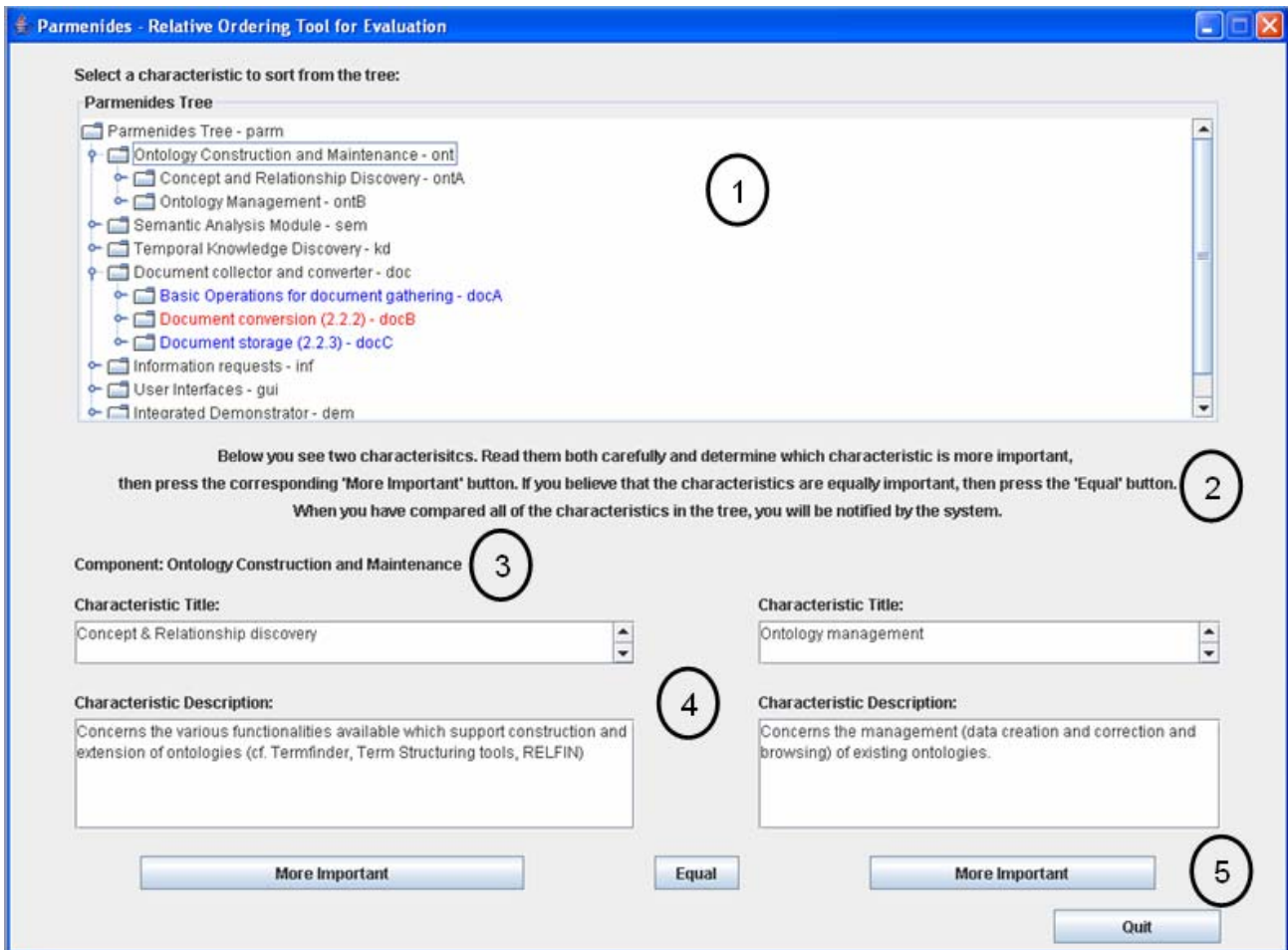


Figure 1: The ROTE Interface

Characteristics which have already been sorted are marked in a different colour. Furthermore, a distinction is made between a tree in which all of the sub-trees have been sorted, and one in which only some of the sub-trees have been sorted, which allows users to quickly and easily see where they still need to work.

In area 2 the user is presented with a brief set of instructions on how to use the system and what to do at a particular point during the interaction.

Area 3 indicates to the user which component of the quality model the sub-tree of characteristics currently being sorted belongs to.

Area 4 of the interface contains two sets of windows - a left-hand set and a right-hand set. Once a sub-tree of characteristics to sort is selected, these windows are filled with the relevant information describing the characteristics and the buttons below them are activated.

The upper window of each of these sets situates the characteristic in the tree. The lower window displays the content of the characteristic i.e. what it is about. The user must read the information in both sets of windows and decide which characteristic is more important to them in the given application. To register their choice, they click on the 'More Important' button beneath the corresponding window set, or on the 'Equal' button if the user feels that the characteristics are equally important. Once a selection is made, a new pair of characteristics from that sub-tree appears and the user must make a selection between those. This continues until all of the

characteristics in a sub-tree have been treated. Once this has happened, the windows are cleared and the buttons deactivated until a new sub-tree of characteristics is selected by the user. The user can re-sort a sub-tree of characteristics that he has already sorted by simply selecting it from the tree again. However, only the results of the most recent sorting are saved.

The 'Quit' button in area 5 allows the user to stop sorting at any time they choose. This is important since quality model trees can be quite large and the user may not want or have time to treat the entire tree in one sitting. The results of the sorting are saved immediately after the sorting of all of the characteristics in the chosen sub-tree is complete. If the user chooses to quit before he has finished sorting a sub-tree of characteristics, the sorting of that sub-tree up to that point will not be saved. The user is made aware of this via a pop-up window and asked by the system if they are sure they want to quit before the system closes.

The results of a ROTE sort are saved in folders in the user's file system, on a per-sub-tree basis. This means that each intermediate node of the quality model is saved as a separate file, containing only the result of the sorting of its immediate children. Furthermore, the files are saved in two formats. The first is a human readable, verbose, format that allows users and evaluators to quickly review and understand the results. The second is a more machine readable format which stores only simple IDs of each characteristic (rather than the verbal

description) so that results can be treated automatically by scripts designed by evaluators.

## 2.2. The Ordering Algorithm

In ROTE, the nodes from within a single sub-tree are compared in a pair-wise fashion, always comparing a new characteristic with the first characteristic from the ordered list in which results are kept. If two characteristics are considered to have the same importance, then they are kept together in the list (in the form of a sub-list), and any future characteristic will only be compared to one of the characteristics from that sub-list, the last one that was added.

Let us take the example of a user sorting a sub-tree with four characteristics {A, B, C, D}. The user is first shown a randomly selected pair {B, D}. They then choose D as more important, so the ordered list is now as in Figure 2.

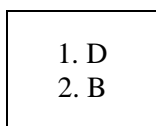


Figure 2: Result of the first comparison.

The user is then asked to compare A (a new characteristic from the sub-tree) and D (the first characteristic in the list). They choose D as more important, so the algorithm goes down to the next item in the list and asks the user to compare A to B. The user decides that they are equally important, so A is added to make a set with B at the second position, as in Figure 3.

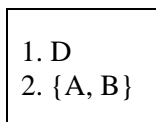


Figure 3: Result of the second comparison.

Finally, the user is asked to compare C, the final characteristic from the original sub-tree, with D. They decide that D is more important, so the user is asked to compare C with A, the first item in the sub-list at position 2. The user decides that A is more important. C is then added to the final position on the list, as in Figure 4.

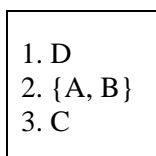


Figure 4: Result of the third comparison.

The user is never asked to compare C to B because they have already said that B and A are equally important, so if the user considered C to be less important than A, they will also consider it to be less important than B.

## 3. ROTE in Practice

As mentioned earlier, ROTE was developed during the Parmenides project in which the current authors were responsible for the design of an evaluation framework to be applied to a large-scale text mining system which was being developed concurrently during the project. Such an evaluation framework for a large and complex system necessarily resulted in a complex quality model containing more than 180 metrics (Underwood & Lisowska, 2006). It was this complexity of the quality model which initially lead us to develop ROTE in order to support the user partners in their work on the determining the evaluation assessment criteria.

ROTE was used without problem by all three users in the Parmenides project. Each user had to sort a total of 82 sets of characteristics comprising the Parmenides quality model.

As indicated above, each intermediate node in the quality model tree represents a characteristic which itself is decomposed into sub-characteristics (and eventually metrics). In other words these sub-characteristics combine to define the parent characteristic. In some cases such sub-characteristics are all of a similar type and the users found the pair-wise comparison between them straightforward. However, in other cases sibling nodes in a sub-tree are of radically different types. For example, the characteristic “Editing in the document repository” is composed of a rather complex characteristic “Interference” (which itself is decomposed into simple characteristics) alongside very simple metrics checking standard operations like “Add” “Delete”, “List” etc. In such cases the users reported that comparisons were hard to make and, in particularly difficult cases, one user resorted to simply assigning equal importance to all the sub-characteristics.

Of the 82 sets of characteristics which were compared, more than half of them were judged to contain characteristics of different relative importance to the user.

Additionally, only 2 of the 82 sets of characteristics were ranked in exactly the same way by all three users, which indicates the diversity of their requirements on the same system. Although the purpose of the exercise was not to make comparisons between the users’ relative ordering, a number of interesting similarities became apparent. It seems that the users preferred to have a system with several core functionalities working very well, and in particular a system that allowed for automation of many of the processes. Furthermore, their rankings seem to indicate that this could be accomplished at the cost of low level convenience functionalities (error reports, extendibility etc), system customization capabilities, and, surprisingly, an interface that was not always user-friendly.

## 4. Conclusions and Future Work

Clearly, with simpler quality models, or for partial evaluations, the use of a tool like ROTE is probably not necessary. However, as the previous section suggests, for more complex quality models there appear to be benefits to using the tool. Our experiences with the ROTE tool thus far have been positive. Users seem to be happy to have a tool that helps them perform an otherwise confusing and boring task in a simple and efficient

manner, and evaluators (in this case ourselves) have found the results easy to read and process.

Before more can be said about the overall usefulness of this tool though, it will need to be tested with other systems of varying complexity. In order to move a step closer to this goal, we have made the tool available for download online at [www.issco.unige.ch/](http://www.issco.unige.ch/) along with instructions on how to structure the input files and run the program. We invite members of the evaluation community to try the tool and provide us with feedback on how it could be improved.

Additionally, we envision that ROTE could be used as a part of a larger evaluation system where users can create an entire quality model 'online', be able to view the whole model and, more importantly, specify all of the parameters gathered during the evaluation for each characteristic. The system would then automatically calculate the overall 'score' for each node in the quality model, all the way from the bottom-most nodes to the root of the tree. With such a general view of the quality model and allowing users/evaluators to specify evaluation results and immediately see what effect they have on the overall evaluation of the tool, evaluators would have the power to test different hypothetical results in order to see how changes to aspects like rating-levels and actual results would effect the overall evaluation. In short, it would allow evaluators to quickly pinpoint the weakest areas of the system and determine what aspects to focus on in order to improve the system for particular sets of user needs. In the case of such a system (which is currently under development at ISSCO/TIM/ETI, University of Geneva), the results from the relative ordering resulting from a ROTE sorting could be fed directly into this system, and conversely, the tree generated in this new tool would be used to populate ROTE.

Finally, although ROTE was developed to help with the evaluation of a text-mining system, it can be applied to any type of user-driven evaluation. Moreover, we believe that the tool can in fact also be applied to any task that requires a relative ordering (in other words a rank ordering) based on subjective judgement. An example of this might be the judging of exams that will need to be situated on a bell-curve in order to assign a final grade.

## 5. References

ISO/IEC (2001). 9126-1 Software engineering – Product quality – Part 1: Quality Model. Geneva, International Organization for Standardization and International Electrotechnical Commission.

Spiliopoulou, M., Rinaldi, F., Black, W. J., Zarri, G. P., Mueller, R. M., Brunzel, M., Theodoulidis, B., Orphanos G., Hess, M., Dowdall, J., McNaught, J., King, M., Persidis, A. & Bernard L. (2004). Coupling Information Extraction and Data Mining for Ontology Learning, in PARMENIDES. In Proceedings of RIAO'2004 (pp 156-170). Le Centre de Hautes Études Internationales d'informatique Documentaire, Paris.

Underwood, N & Lisowska, A (2006), The Evolution of an Evaluation Framework for a Text-Mining System. In Proceedings of the Fifth International Conference

on Language resources and Evaluation, (LREC 2006). (in press).

## 6. Acknowledgements

ROTE was conceived by Maghi King, Nancy Underwood and Agnes Lisowska and implemented by Agnes Lisowska and Philippe Baudrion

The work described here was funded by the Swiss Office Fédéral de l'Éducation et de la Science (OFES) as part of our participation in the Parmenides project No: IST-2001-39023