The Implementation of an MCDM Tool for the Acquisition of Military Equipment

Frédéric Hallot, Philippe De Beir, Hugo Pastijn
Royal Military Academy
Renaissance Avenue 30
1000 Brussels (Belgium)

frederic.hallot@rma.ac.be
hugo.pastijn@rma.ac.be

ABSTRACT

From our experience as well in teaching as in consultancy at the Royal Military Academy we learned that previously existing MCDM software did not fit well acquisition projects of military equipments. To solve this problem, we created a team to develop the "MCDMTool" which is an implementation of the combined use of two different types of outranking methods PROMETHEE and ORESTE. The choice of all technical parameters is assisted by interactive modules. The capabilities of this software are demonstrated here in the framework of acquisition of military equipment.

1.   INTRODUCTION

We are used at the Royal Military Academy to use PROMETHEE or ORESTE outranking methods not only as part of teaching OR (Operation Research) but also as consultants when we have to help the external decision maker facing multi-criteria problems.

Those two methods have an opposite (complementary) approach as PROMETHEE considers that all evaluations of criteria are treated as purely quantitative where ORESTE only allows rankings (qualitative criteria).

The reality of multi-criteria problems though is that we mostly face a mix of criteria, where some of them are clearly quantitative as some others are more qualitative and thus better adapted to ranking than to numerical evaluation.

2.   EXPERIENCE AT RMA

Our experience at RMA, especially when we were asked to help as consultant in the elaboration of projects for acquisition of military equipment in the Belgian army helped us to point out several problems with the software we had at our disposal.

2.1.   *Problems with DOS versions*

Until we started developing ourselves MCDMTool (see later), we used two independent old DOS (Disk Operating System), one called promcalc.exe (ULB) when we had to compute PROMETHEE problems, and the other called oreste.exe (RMA) when facing ORESTE problems[1].

---

[1] *A third helper DOS application called saaty2.com was also sometimes needed when we had to convert rankings into numerical values*

Report Documentation Page

1. REPORT DATE **OCT 2009**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Implementation of an MCDM Tool for the Acquisition of Military Equipment	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Royal Military Academy Renaissance Avenue 30 1000 Brussels (Belgium)**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA562565. RTO-MP-SAS-080 Decision Support Methodologies for Acquisition of Military Equipment (Methodologies d'aide a la decision pour l'acquisition d' equipements militaires). Meeting Proceedings of Systems Analysis and Studies Panel (SAS) Specialists Meeting held in Brussels, Belgium on 22 and 23 October 2009., The original document contains color images.

14. ABSTRACT
From our experience as well in teaching as in consultancy at the Royal Military Academy we learned that previously existing MCDM software did not fit well acquisition projects of military equipments. To solve this problem, we created a team to develop the MCDMTool which is an implementation of the combined use of two different types of outranking methods PROMETHEE and ORESTE. The choice of all technical parameters is assisted by interactive modules. The capabilities of this software are demonstrated here in the framework of acquisition of military equipment.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**10**	

2.1.1. Two separate implementations

As already stated, the Software we used at the RMA were two old independent DOS implementation, one for applying ORESTE and one for applying PROMETHEE.

This situation was not very comfortable and lead to some problems. Indeed most of the time the reality of an MCDM problem is not as simple as having all its criteria being either purely qualitative or purely quantitative. Instead we mostly face problems where some criteria are qualitative when others are quantitative.

Choosing a priori one of both softwares implied the necessity for the evaluator to artificially cast some criteria in a form that doesn't really fit its reality.

In other words the problem had to be reduced to a purely quantitative problem if PROMETHEE was chosen or a purely qualitative problem if ORESTE was chosen.
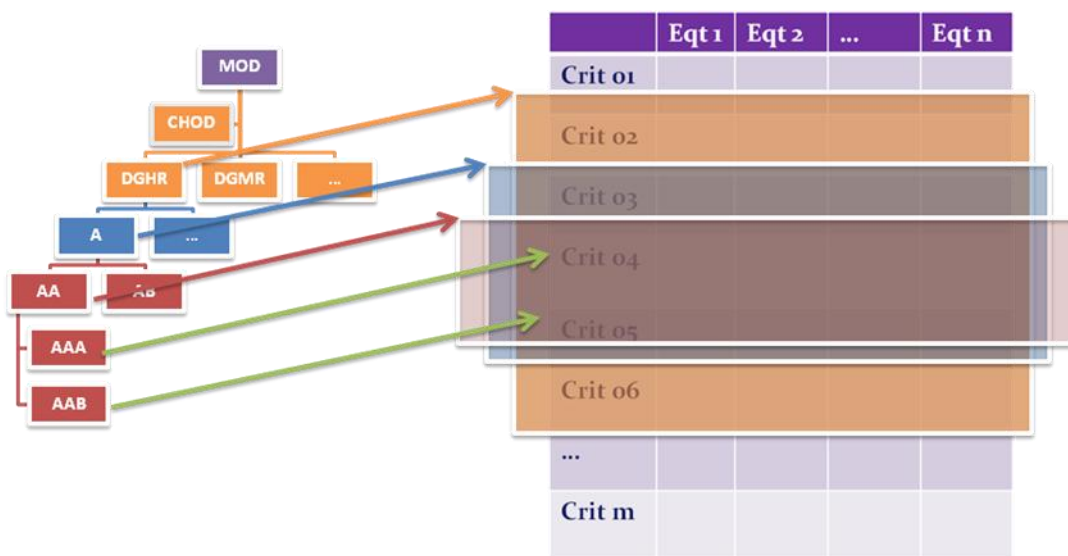
2.1.2. MCDM Problem size

The supported size for MCDM problem in both implementations was very limited in supporting huge number of actions or criteria.

In oreste.exe for example limitations were 24 criteria and 42 actions. But even far before reaching these limitations, the software became difficult to use because of the screen capacity possibilities that did not allow getting a complete view on the problem.

2.1.3. MCDM Problem structure

In both versions, MCDM problems had to be structured in the form of a simple rectangular matrix of actions and criteria.

This way of having to organize all criteria in a flat list did not match the realm of military equipment acquisitions as evaluation of subset of the criteria are often spread down into the hierarchy (See Figure 1).

Figure 1: Mismatch between Mil Hierarchy and standard MCDM projects

2.1.4. Compatibility issues

Both of these old DOS implementations began to have Windows compatibility problems since Windows XP, and these problems grow with the newer versions of MS Windows (Vista and in Widows 7 (since October 22 2009)).

Furthermore, as being DOS implementations, they were also not compatible with any non Microsoft Operating Systems as MacOS, Linux, Solaris, …

3. MCDMTOOL

3.1. *Team*

In order to solve all the problems encountered during our experiences with both DOS versions, we decided to gather an independent work team of IT and OR specialists (F. Hallot, P. De Beir and H. Pastijn) in order to implement a new version that has been called MCDMTool[2].

3.2. *Objectives*

The next part of this paper will enumerate the principal objectives our team wanted to achieve with this new software and will illustrate the results with screenshots.

3.2.1. A Graphical User Interface (GUI)

Our first important goal was to provide an application with a rich windowed Graphical User Interface (GUI) for giving a great user experience compared to the possibilities of DOS versions.
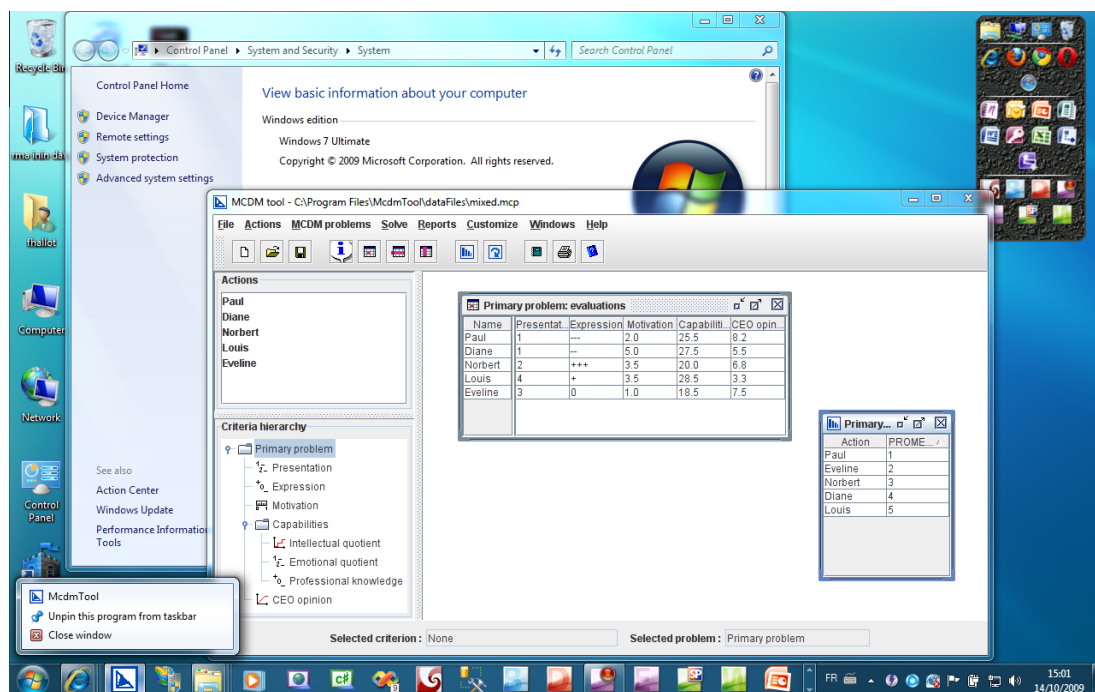

Figure 2: MCDMTool GUI on Windows 7

---

[2] *The people of the team worked during their spare-time and with their own HW (Hardware) and SW (Software), and although there exists no official Start-up yet, MCDMTool is not a project of the RMA, and belongs completely to its inceptors.*

3.2.2. Better Compatibility

MCDMTool was implemented using Java. Although this seems to be a purely technological aspect, this choice was of uttermost importance to ensure portability.
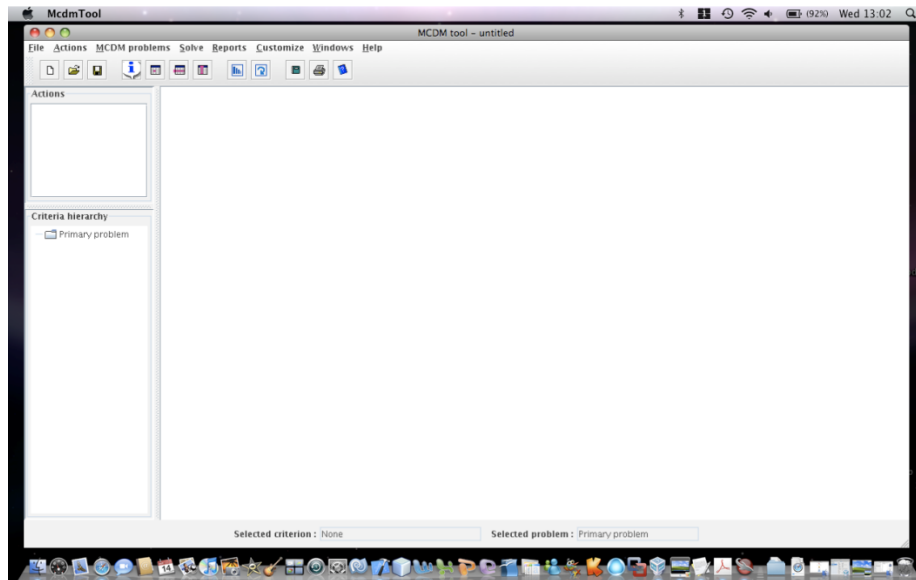

Figure 3: MCDMTool on MacOS X

Indeed, MCDMTool can be used on all operating systems able to run the JVM (Java Virtual Machine) and thus on quite all versions of windows (95, 98, 2000, XP, Vista and Windows 7 (see Figure 2)) but also on MacOS X (see Figure 3), Linux, Solaris and Unix.

3.2.3. ORESTE and PROMETHEE integration

Our next main goal was to integrate both methodologies into our single application.

We could have implemented both methodologies completely separately. This would already have facilitated the life of the user which would not have to preselect one of both methods to start using the corresponding software. But as we already stated, the reality of most MCDM problems, and especially in the field of military equipment acquisition, necessitates frequently the use of a mix of quantitative and qualitative criteria.
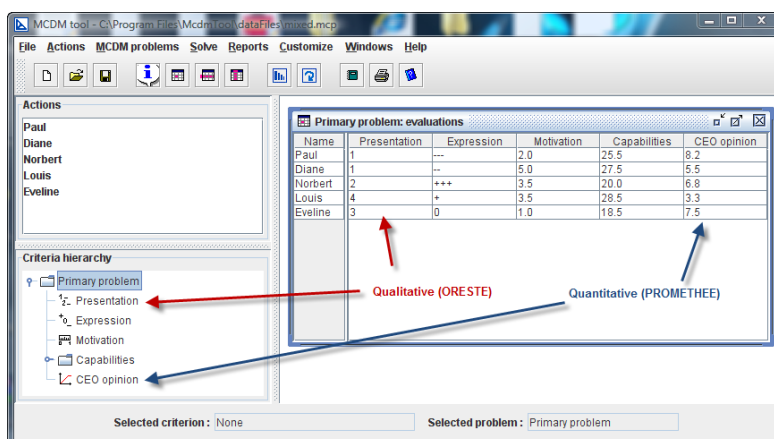

Figure 4: Integration of PROMETHEE and ORESTE in MCDMTool

So our goal was even more ambitious than only implementing both methodologies separately. We wanted to give the user to intertwine both types of criteria together in a single project. This would allow the assessor to evaluate the candidates on each criterion in a way close to its inherent nature (See Figure 4).

We even added two predefined types that we called Appreciation[3] (extension qualitative type - ORESTE) and Cardinal (extensions of quantitative type – PROMETHEE) but that augments once again the user experience when having to evaluate those types of criteria (See Figure 5).
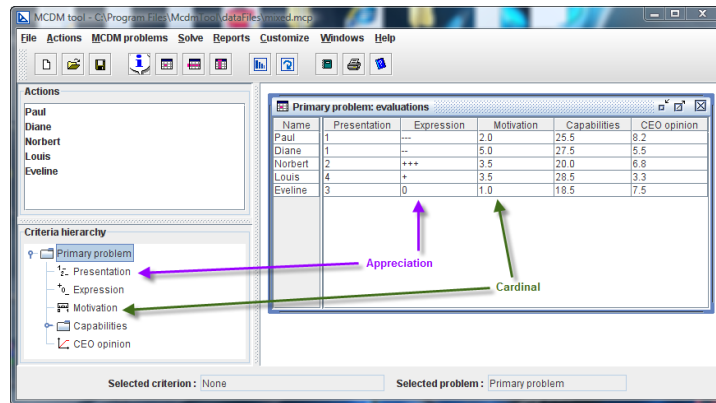

Figure 5: Supplementary criterion types in MCDMTool

This integration allows the users to delay the choice of the methodology chosen for computing the results until the decision time, and allow even to analyse the results with both methodologies (See Figure 6).
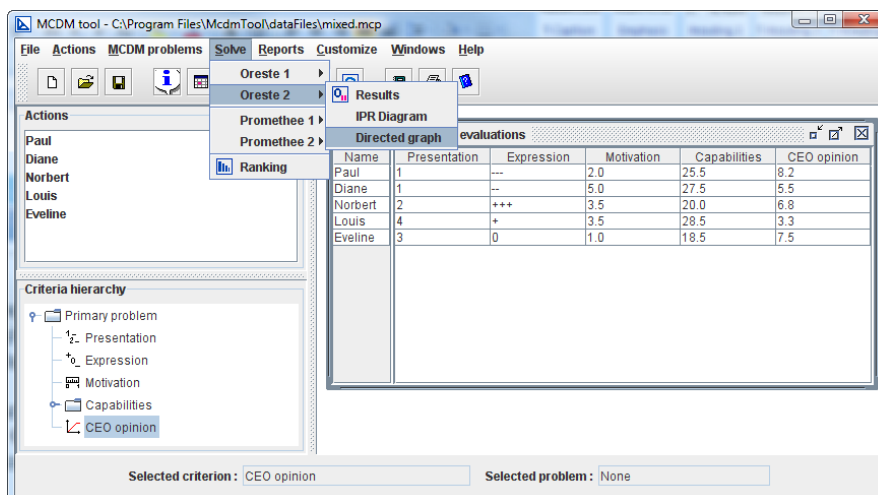

Figure 6: ORESTE and PROMETHEE can analyse a same MCDMTool project

This integration, though required implementing a lot of helper tools for the conversion of ORESTE criterion to PROMETHEE ones and vice-versa (See Figure 7), as well as for computing ranks from weights and vice-versa (See Figure 8).

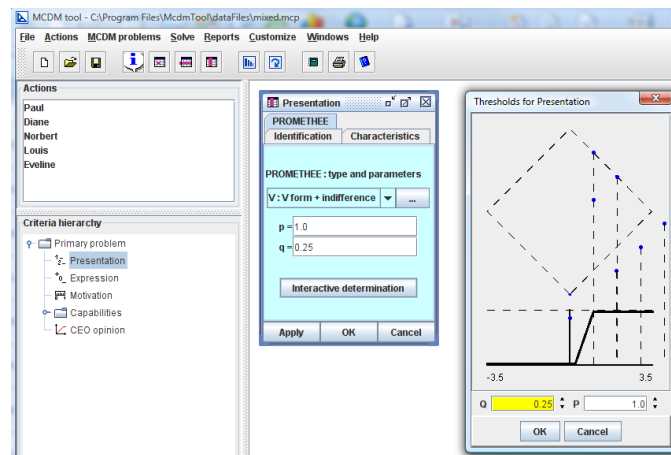[3] *To be renamed « Assessment » in the next version.*

Figure 7: Helper tools for integrating an ORESTE criterion in PROMETHEE.
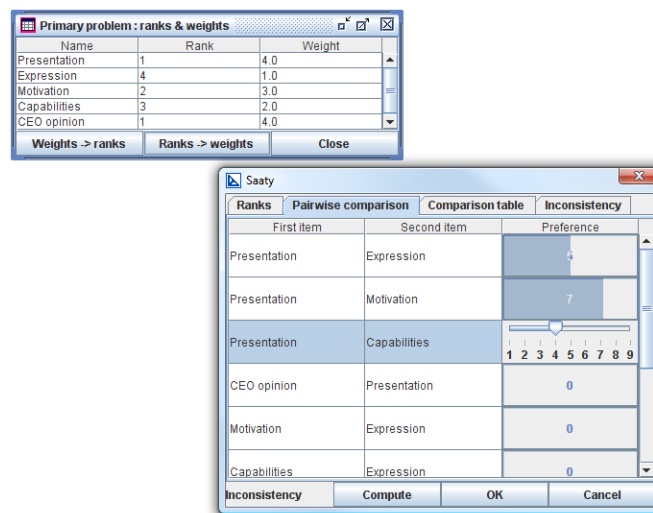

Figure 8: Ranks – Weight Conversion helper tools

3.2.4. Facilitation of Group Decision Making

As already shown in Figure 1, the acquisition process of military equipment is teamwork involving several people at several levels of the military hierarchy.

As we analyse the real functioning of such acquisition projects, some very high end criteria are defined at the top level. Those criteria are given for evaluation to different subordinate levels. Most of the time the subordinate level can't evaluate the criterion directly and considers it in turn as an MCDM problem per se, that can be further decomposed in criteria (sub-criteria). Those new sub-criteria are then sent down into the hierarchy to the competent evaluation level.

This process of further refining criteria by decomposing them in sub-criteria that are sent a level down into the hierarchy can happen at several levels of the hierarchy, until we reach "leaf" nodes where the evaluations effectively happen. Results are then sent back to the upper level where they will be integrated in the corresponding MCDM project that will deliver results that will in turn be returned at the upper level for further integration until the "root" project is reached. Once every criterion of the "root" project has been received, the final results can be computed in order to help the top management to take the appropriate decision (See Figure 9).
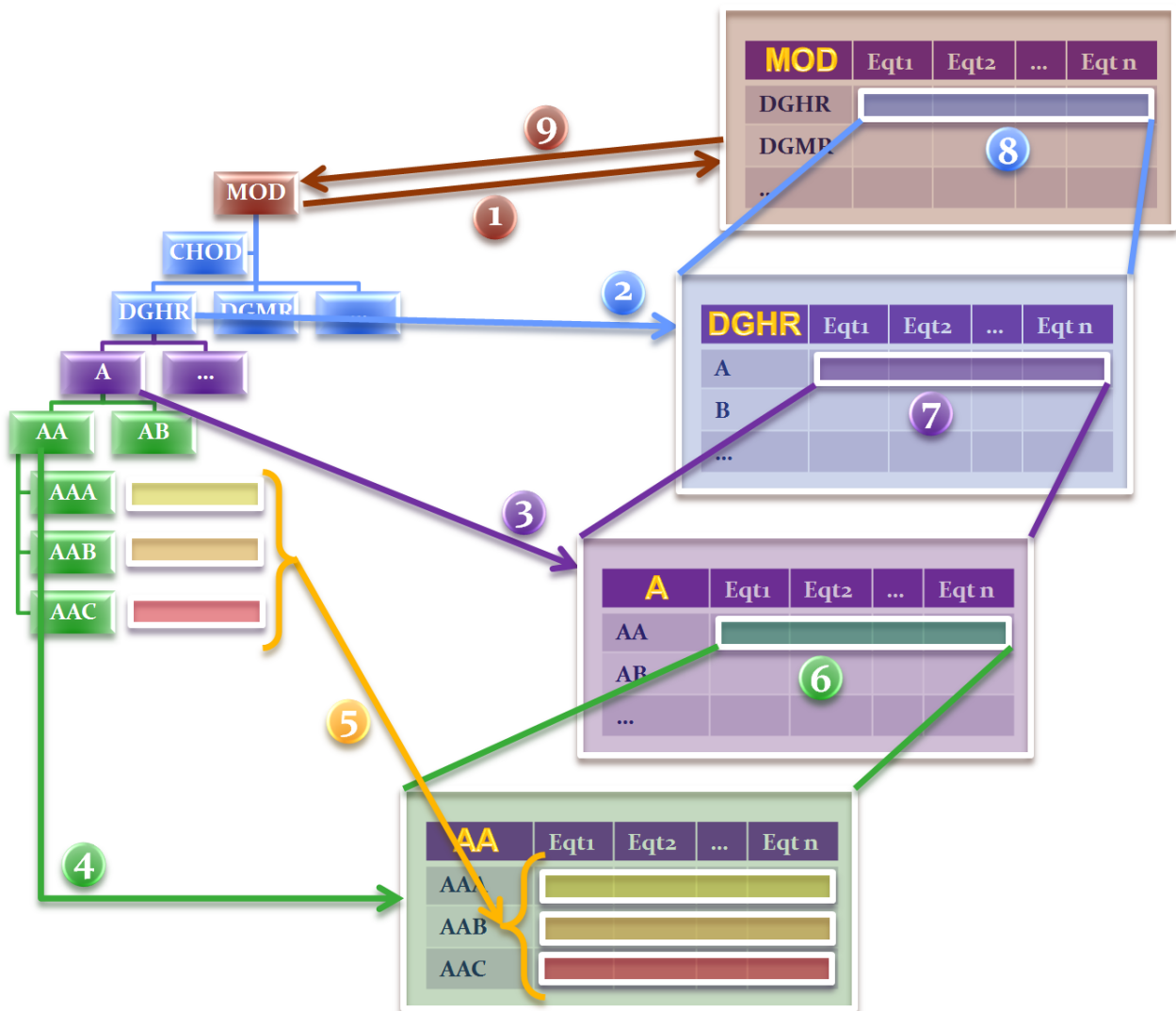
**Figure 9: The effective process when using MCDM in a hierarchy**

This procedure is quite tedious, error prone as the results sent from one level to the upper level must be re-encoded and if some error should be corrected at a lower level, no automatic correction would happen at the upper level. This makes the procedure quite long as every evaluation has to bubble up the hierarchy before results can be computed for the decision maker. And if for some reasons the decision maker is not happy about the results, because they don't seem to be correct and require further refinements of the evaluation process, the whole procedure has to start again.

3.2.4.1. Introducing a new type of criterion: MCDMProblem

So even more important than integrating PROMETHEE and ORESTE, our top level goal when incepting the MCDMTool project was to transform the pure rectangular MCDM projects in a form that would better suit the reality of decision making in big companies (as the Belgian Defence).

The solution was not that difficult to find. Introducing a new type of criteria called MCDMProblem, that can be further decomposed into sub-criteria, would fix the problem. And in fact, it was just a computer implementation of a manual practice inside our institution. The MCDM project would remain a two dimensional plan, but a bit more complex as one of its axes would not be a flat list anymore but a hierarchical tree.
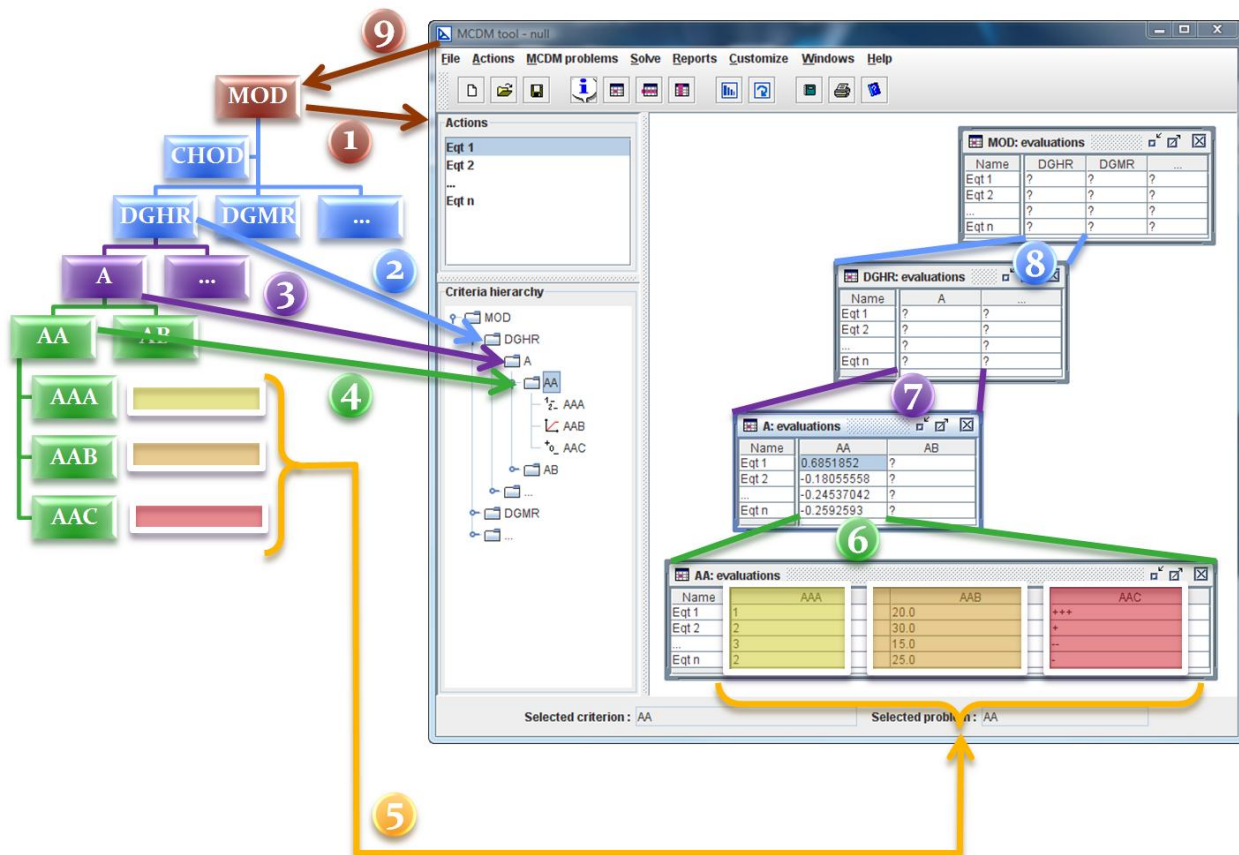
Figure 10: MCDM Criteria hierarchy & evaluation propagation through MCDMProblem type

When all sub-criteria are evaluated, the parent MCDMProblem can be "solved" and its results can be considered as the criterion evaluation at the upper level (See Figure 10). With MCDMTool only one MCDM project has to be created now. It can be further expanded by the lower levels of the hierarchy and when evaluations are entered at "leaf" nodes, they are immediately integrated up into the higher level, at least if all integration parameters have correctly been set (See Figure 10).

3.2.4.2. Provide better reusability

Reusability is also an important part of Group Decision Making. It was an initial choice that MCDMTool project files should be readable and editable without having to use MCDMTool. Because the data composing an MCDM project is highly structured, it seemed obvious to us that XML (eXtensible Markup Language) would be the perfect candidate for saving project data.

This format allows predefining templates of criteria for example, when a set of criteria have to be used regularly in different MCDM projects.

It was also important for us to foresee the possibility to store MCDM project data in relational databases. This possibility exists in MCDMTool but is not available directly from the GUI. In the case one wants to rely on database for storing project information, some light modifications have to be done within the code before recompiling, but this option is not accessible in the standard software.

3.2.5. Efficiency features

A lot of features were in our original goals and have been incorporated inside MCDMTool. Nevertheless they are general features and not especially targeted for the military equipment acquisition type of problem. Hence we will only cite a non-exhaustive list of them without explaining them further or illustrating them with screenshots, but they remain important improvements of what was at our disposal when we had to work with those DOS versions.

MCDMTool comes thus also with

- A wizard that allows computing automatically a ranking without having to adjust all parameters when all evaluations are provided.

- The possibility to rule out actions or criteria from computing without having to delete them which would as side effect imply losing their evaluations.

- A reporting tool (generating .html file with results, graphs and diagrams)

- Graphical tools for analysing results with graphs and IPR diagrams (not only for ORESTE but also for PROMETHEE)

- All kinds of wizards in order to graphically help during determination of all the sorts of parameters

- Colorization (customizable) from all ORESTE and PROMETHEE windows in two different colours in order to help the users knowing in which context he sees results.

- Possibility to consult all intermediate results

- Import tools for both types of old version project files (promcalc.exe and oreste.exe)

4. CONCLUSIONS

Two big steps forward have already been made in MCDMTool, towards a better support tool for decision making concerning the acquisition of military equipment.

Indeed by accepting the mix of qualitative and quantitative criteria in one single problem, MCDMTool helps its users to build their projects in a more natural way without forcing them to artificially cast some of their qualitative criteria to quantitative ones or the contrary. Now, every criterion can be evaluated according to its type which eases a lot the life of the assessor.

The most interesting progress of MCDMTool approach though is the transformation of the MCDM problems from their traditional flat list structure of criteria to a tree structure one by adding recursion to MCDM problems by allowing a criterion to be qualitative (ORESTE), quantitative (PROMETHEE) or an MCDM problem per se (that can further be decomposed in sub-criteria). This new approach fits much more big companies like the Belgian Defence where the final decision has to be taken at a very high level but where the evaluation of the criteria is delegated to several services in a hierarchical fashion.

Although not an intellectual property of the RMA, MCDMTool is (and will always remain) freely used at RMA-TMWA during teaching and practical works of OR courses as well as in order to answer some consultancy missions.

5. THE FUTURE OF MCDMTOOL

Despite its new hierarchical approach, MCDMTool is still not satisfactory supporting multiple users, which is still an obstacle to true collaborative (group) decision making.

The next step in the evolution of MCDMTool in order to provide an improved tool for the acquisition of military equipments will be to integrate a user management module.

In a first instance this module should allow a project manager to define several evaluators with adapted access rights and to partition the project by assigning each criterion to one of the evaluators. This evolution would not need any scientific justifications as the only change in the software would be to allow several users to accomplish the same work that had to be done previously by a single actor by distributing evaluation responsibilities.

In a second time, it would be interesting to allow each evaluator to evaluate every action on every criterion, like in a jury. This evolution would however need a scientific justification as a third dimension would be added to the problem transforming it in a kind of data cube. An extra level of aggregation would become necessary and hence would have to be scientifically explained.