HAL open science

Classification procedures for software evaluation

Sophie David, Rachel Panckhurst, Lisa Whistlecroft, Muriel Amar

To cite this version:

Sophie David, Rachel Panckhurst, Lisa Whistlecroft, Muriel Amar. Classification procedures for software evaluation. LREC 2008, Jun 2008, Marrakech, Morocco. pp.Publication électronique. hal-00285418v2

HAL Id: hal-00285418

https://hal.science/hal-00285418v2

Submitted on 16 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Classification procedures for software evaluation.

Published with the permission of ELRA. This paper was published within the proceedings of the LREC 2008 Conference.
© 2008 ELRA - European Language Resources Association.
All rights reserved.

Muriel Amar[1], Sophie David[2], Rachel Panckhurst[3], Lisa Whistlecroft[4]

[1] Urfist de Paris - École nationale des Chartes, 17, rue des Bernardins, 75005 Paris, France,
[2] CNRS UMR 7114, MoDyCo, Université Paris 10, bât. L-R12A,
200, av. de la République, 92001 Nanterre cedex,France,
[3] Praxiling UMR 5267 CNRS, Université Paul-Valéry Montpellier 3, 34199 Montpellier cedex 5, France,
[4] PALATINE, The Great Hall, Lancaster University, Lancaster, LA1 4YW, UK.
E-mail: muriel.amar@enc.sorbonne.fr, sophie.david@u-paris10.fr,
rachel.panckhurst@univ-montp3.fr, Lisa.Whistlecroft@lancaster.ac.uk

Abstract

We outline a methodological classification for evaluation approaches of software in general. This classification was initiated partly owing to involvement in a biennial European competition (the European Academic Software Award, EASA) which was held for over a decade. The evaluation grid used in EASA gradually became obsolete and inappropriate in recent years, and therefore needed to be revised. In order to do this, it was important to situate the competition in relation to other software evaluation procedures. A methodological perspective for the classification is adopted rather than a conceptual one, since a number of difficulties arise with the latter. We focus on three main questions: what to evaluate? how to evaluate? and who evaluates? The classification is therefore hybrid: it allows one to account for the most common evaluation approaches and is also an observatory. Two main approaches are differentiated: system and usage. We conclude that any evaluation always constructs its own object, and the objects to be evaluated only partially determine the evaluation which can be applied to them. Generally speaking, this allows one to begin apprehending what type of knowledge is objectified when one or another approach is chosen.

Keywords: Evaluation, Methodological classification, Software, Competitions, TREC, MUC, EASA, Epistemology.

1. Introduction

Over approximately the past twenty years, the domain of evaluation has attempted to become an independent field, through international conferences (*e.g.* TREC, Text Retrieval Conference; MUC, *Message Understanding Conference*; LREC, *Language Resources and Evaluation Conference*), competitions (*e.g.* EASA, *European Academic Software Award* (Panckhurst *et al.* 2004)), publications (*e.g.* Sparck-Jones & Gallier, 1996; Chaudiron, 2004), and international agencies (*e.g.* ELDA, a French *Agency for evaluation & distribution of linguistic resources*; NIST *National Institute of Standards and Technology*). The production of new software devices has increased; it has mainly emerged in response to professional demand, *e.g.* in natural language processing or engineering: spelling and grammar checkers, tokenisers, machine translation systems, voice recognisers, etc.; but technological developments also emerged early on in information retrieval (IR) (Chaudiron, 2004). At the same time, social demands have made it necessary to account for the appropriateness of this research, and evaluation procedures have been developed, thereby extending longstanding traditions of evaluation principles for software devices both in linguistics and IR (*cf.* for the first evaluation reports in linguistics and data processing, Bar-Hillel, 1960, ALPAC report, 1966, and for a state of the art historical perspective, Cori *et al.*, 2002, Cori & Léon 2002). Much work has been done, and evaluation approaches can now be studied as such: i) procedures can be classified, owing to their relative diversity, which is now well-documented; ii) the way to characterise objects to be evaluated can be queried.

This work includes several aims: 1) produce a classification, which is methodological in nature; 2) focus on the complex nature of all evaluation approaches; 3) start stipulating what type of knowledge is objectified throughout all evaluation approaches.

First, we situate the context of our study (the EASA competition, which partly provided the initial impetus at the onset of our research) and the issues at stake (§ 2.). Then we defend the relevance of a methodological classification for evaluation approaches of software in general (§ 3.), before proposing different elements to produce a classification of the most common evaluation approaches (§ 4.). We finally discuss the two fundamental types of approach which emerge from this classification (§ 5.).

2. The context

2.1. The EASA Competition

The European Academic Software Award (EASA) was initiated in 1994 and the last competition was held in 2004. It was a biennial competition which was organised by the European Knowledge Media Association (EKMA). Academics and students were able to submit software they had

developed which was then evaluated by a team of European jurors. After an expert juror evaluation process of 150 to 200 submissions, 30 to 35 items were selected to proceed to the third and final stage. The finalists' submissions were evaluated once more and 10 prizes were then allocated to the winners. Over the years some aspects of the evaluation process and criteria became inappropriate or obsolete, due to several factors:

- a very wide scope of entries (in later years, EASA implicitly became a competition including not only software but also virtual learning environments (VLEs) and pedagogical innovations using VLEs);
- technical improvements became standard (it was not relevant to evaluate these as they no longer allowed appropriate differentiation);
- some of the questions in the evaluation grid became spurious and/or ambiguous, etc.

Three of the authors were therefore commissioned by EKMA to conduct a revision of the whole procedure, but in order to do so, they realised that EASA needed to be situated in relation to other software evaluation procedures, namely: to improve comparisons between competitions; to put emphasis on EASA's original elements and to confront solutions adopted within other competitions in order to improve the weak points of the EASA procedure.
This research is partly based on previous work (*e.g.*, the distinctions proposed by Sparck-Jones & Gallier, 1996), but it also integrates other procedures, *e.g.*, work on usage, or the EASA competition, which includes several original elements. Following the initial impetus of this research (to improve the procedure of the EASA competition), the objects subjected to evaluation procedures that we want to characterise remain systems in a broad sense: software, VLEs, etc. Static resources (corpora, databases, etc.) are not considered in this paper[1].

2.2. Issues at stake

It appears to us that the field of evaluation was initially posited in a problematic way, by considering that it could be described conceptually as a discipline, and by positioning itself as an autonomous science with its own concepts, methods and rules (*e.g.*, Ellis, 1992 quoted by Chaudiron, 2004; Sparck-Jones & Gallier, 1996). Needs for evaluation, from an industrial or research perspective, the necessity for rigour and systematism which accompany these projects, and assets in terms of results do not imply that evaluation should be considered an autonomous science as such. Evaluation is a methodological step of every system, for every project. It seems that this conceptual status is often *a posteriori* reconsidered. Our current research goes in the opposite direction to much former work, by establishing methodological distinctions (*cf.* (§ 3.)).
In addition, in a world where evaluation has become increasingly important, and in which its results have consequences at different levels (professional recognition of work, leading to research funding, commercialisation of products, etc.), it is important to characterise the context of an evaluation and the adopted procedures at their best. One often observes that evaluation is ultimately founded on measures (*cf.* Sparck-Jones & Gallier, 1996, p. 20-21, measures for evaluation of tokenisers, Adda *et al.*, 2000). Even though it is trivial to state that what we measure is measurable, it is much less trivial to discuss the *meaning* of what is measured. The measure is no longer a simple measure, it becomes an *indicator*. The measure then shifts from its calculus space — its context of production — to another space — its context of interpretation, in which indices not belonging to the calculus variables are used, and which possibly summon yet other ones. Different examples illustrate this point:

1. *Temperature recordings*: the same temperature is interpreted differently according to the season, the geographical situation, etc.
2. In companies, the *absenteeism rate* is admittedly measurable, but its meaning is not fixed for all companies, nor is it for a specific company, because it always depends on a particular context and can alter over time, owing to: relational deficiencies between managers and employees, within a specific professional group, problems about hygiene or security, anxiogene social pressure, etc. Absenteeism as such is easy to measure; however, the interpretation of this as an indicator is a much more complex question.
3. The *number of visitors* at the BPI (*Bibliothèque Publique d'Information*, the biggest library in Paris): the BPI registers 6,000 entries per day, except on Sundays, where this number drops to 4,000 entries (figures are approximate). As there is a limit of 2,000 people in the library at one time, the library "fills" 3 times every day, but only twice on Sundays, when the queue is surprisingly lengthy. Another indicator is necessary in this case: the duration of the visit, which is longer on Sundays. Which indicator is the most reliable to account for what happens on Sundays? It depends on the question: accounting for a phenomenon (a social sciences approach) or measuring the conformity of an event (even a social event) to a law (a reference, a scientific approach, etc.).

The indicator (built on a measure) is thus necessarily interpreted by some indices which are not part of the calculated elements. The dimension of this interpretation requires another framework, distinct from a theoretical framework on which the measure is founded (for example physical measures *vs.* climatological interpretations). Finally, it is important to underline that what is at stake in an evaluation is crucial when one knows the ability of humans (including researchers) to adapt easily to evaluation procedures. If, on the one hand, clarification must be transparent, on the other hand one must be aware of the impact that procedures can have on designing systems. In other words, differentiate between what a community is able to build in terms of objects, and what a community of experts is able to evaluate. And it is unfortunate that, for non-scientific reasons, the constructed objects suffer from constructed evaluation

[1] It is of course clear that more precise observation of the procedures that the community has proposed for these types of resources would be an asset.

procedures (*e.g.* in one of the past TREC competitions, answers were limited to 50 characters maximum, Lavenus & Lapalme, 2002).
More fundamentally, our research tries to characterise the type of knowledge which can be addressed when we posit evaluation procedures (in a similar way to Pariente's work (1973) about conceptual knowledge). This aim exceeds the framework stipulated in the current paper by far. However, the classification we propose is a first step. It begins to establish that:

1. every evaluation always builds its own object of study;
2. objects to be evaluated partially determine the procedure which can be applied to them. This is what one can explicitly perceive in the examples associated with the classification, where the same object can be evaluated according to different procedures.

3. A methodological classification

The classification proposed below (§ 4.) relies on a methodological approach to evaluation. Classifying approaches with different aims does not allow the development of conceptual observatories as such. This is due to the fact that: (i) evaluation aims may include differing scientific, social, financial, etc. considerations (Habermas' (1973) definition of a practice is more relevant here than that of a conceptual domain); (ii) evaluation is applied to fundamentally multidisciplinary objects (in computer sciences, linguistics, IR, communication, learning, etc.). If these objects were conceptually characterised, one would have, at best, a set of concepts elaborated in all implied disciplines, but this set could not form an integrated theory. Furthermore, articulating the conceptual framework, which produces the data, with the conceptual framework which produces their interpretation (see § 2.2. for examples) would become necessary, which is not a trivial problem; (iii) outside the classification framework, can a specific evaluation be linked with conceptual knowledge? If this is the case, concepts and theories need to be determined; nothing of the kind has been convincingly demonstrated so far, including specific elements of the domain: for instance *glassbox/blackbox* are not concepts that belong to a particular theory (or theories) of evaluation but are only methodological notions. Though it is proposed by many authors (*e.g.* Sparck-Jones & Gallier, 1996: "this section introduces some basic, general evaluation concepts" (p. 19); "The main problem in evaluation is finding measures, *i.e.* concepts which are both instantiations of generic notions and are operable as measures" (p. 20); or Chaudiron 2004, who extends Ellis' work (1992), by using the term "paradigm"; see also Chaudiron & Mustafa el Hadi, 2007, for usage of this term). It is not sufficient to name a notion a "concept" for it to really become one. The concept would have to be integrated into a conceptual network, and be defined according to a "study object"; (iv) the multi-disciplinarity of the objects to be evaluated prevents one from giving a stabilised definition of what could be a "study object" of the evaluation, in this case as a conceptual discipline. More precisely, a definition of a "study object" as such does not exist, in the way it does in linguistics for instance ("characterise 'language' in relation to 'non-language'", according to Milner's (1989) research program); or even in information sciences, where the study object is the study of the "process of research and exploitation of the intentional information" (or "communicated knowledge", which differs from "news-information", "data-information" "knowledge-information", Fondin 2006). For these reasons, could a theory exist, which dominates all other theories implied in both the building of measures and of interpretations? Facing this epistemological issue, we have chosen a stance which solely posits "methodological distinctions". As these "methodological distinctions" can be applied to all approaches, they can be compared. And, as is shown below, methodological questioning allows the construction of an observatory (Milner, 1989). Three questions are sufficient for classifying the most common approaches (but not for describing each one in detail, but this is not our purpose). These 3 questions are: What to evaluate? How to evaluate? Who evaluates?

4. Elements of the classification

We now review the different elements and sub-elements that we have posited. The general classification appears in Table 1 (see Appendix).

4.1. What to evaluate?

4.1.1. Objects to be evaluated

This indicates whether the evaluation primarily takes into account the software, or primarily considers usage:

1. The objects which are evaluated are items of *software*, isolated from their context of use. They may consist of one or several items of software. The latter may consist of the same or different types of software.
2. Another method is centred on *usage*. The item of software is evaluated in its context of use. The evaluation must therefore take into account many other factors, which form a complex device (purpose, users, expectations, etc.). Research conducted by Le Marec (2004) on evaluation in the context of museums is an example. She illustrates how computerised information points in museums are used, and that they are only one factor among many which form a complex device of institutional communication, including: expectations, itineraries, pieces of information appearing near the information points, etc. In actual fact, it may not be the computerised information point as such, which should be evaluated, she stresses, but rather the situation as a whole.

4.1.2. Access

The evaluator engages with different elements of the software depending on the type of access. Two methods appear:

1. The *glassbox* method implies that the evaluator has access to the whole computing process (structure, algorithms, programming). It includes detailed evaluation, and is often accompanied with measures of intrinsic performance of the software. Reasons and causes of errors/bugs are investigated from a computer programming perspective. Several key stages are analysed and the results influence later development;

consequences are both financial and human. The developer often conducts this sort of evaluation (Falkedal, 1998).

2. The *blackbox* method focuses solely on input and output. The evaluator does not have access to any details of the computer process, which remains a black box. This method is generally used when there is intellectual or commercial copyright, and is often used in competitions (*e.g.*, TREC, MUC, EASA, etc.).

4.2. How to evaluate?

4.2.1. Object distribution: individual or comparative

This refers to the evaluation of multiple items.

1. The items of software are evaluated *one by one*. The evaluation procedure (which may be fairly detailed) is applied to each item of software individually. This is the most common method used in competitions (*e.g.*, TREC, MUC, EASA, etc.).
2. The items of software are evaluated *comparatively*, together. A common point of view is established, allowing for similarities/differences. This perspective is not detailed and is always *ad hoc*, since it is constructed on the basis of participating items of software (considered in a sense as tokens but not as the instance of a type). Compared to 1), only a small number of items of software may be evaluated. This method is usually used in order to create connections between software developers. For instance, this was the initial framework chosen for the evaluation project of information extracting devices (funded by the Agence universitaire pour la francophonie, AUF, Amar & David, 2001).

4.2.2. Resources

This aspect refers to the means used during the evaluation :

1. Referentials are used when stable, consensual and normed knowledge exists, or when expected results can be stated in advance; referentials give a form of external calibration (for instance spelling and grammatical rules for a spelling and grammar checker). The results produced by the software are considered to be correct or incorrect. This method is often used when ranking of software is required, since the referentials are used to make comparisons between items of competing software.
2. No referentials are used when stable and normed knowledge does not exist, or when expected results cannot be stated in advance. This is often the case for situations which are more or less consensual or when one focuses on needs which can change according to differing practice and context. Instructional software may be typical of this sort of approach, but also software for indexing (Amar & David, 2001) or automatic summarisers, for instance, in which needs change according to differing practice & context (bibliographical summaries, those produced in academia, firms, etc., Abbou 2000; and for a review Minel, 2004), or machine translation systems (different people, such as engineers, experts, academics, etc. have different needs: translating a word, a sentence, an article, etc.; King & Falkedal 1990; Nübel & Seewald, 1998).

4.2.3. Measures

Quantitative vs. qualitative methods can be applied.

1. In quantitative methods, a mark is attributed to evaluated aspects (via sets of tests/questions about content, interface ergonomics, etc.). Marks are usually associated with true/false answers or check-boxes on a grid. Quantitative methods are often used in competitions since marks are then ranked. Gold-standard methods can be included here: the software is measured against a given gold standard (which is established from a set of expected answers).
2. A qualitative method refers to a particular issue; in this instance, a methodology and a questionnaire are often used. The result is usually a report including recommendations. This does not mean that all aspects are excluded from any sort of measure, but simply that the measure is never seen to be the final result of the evaluation (Le Marec, 2004).

4.2.4. Evaluation distribution

This is where we consider the number of evaluations and the ways in which the evaluators work.

1. Single: The software may be evaluated by one evaluator.
2. Aggregated: The software may be evaluated by several evaluators and the evaluation results of the several evaluations combined.
3. Collective: The software may be evaluated by several evaluators who produce a single, agreed or negotiated evaluation.

4.3. Who evaluates?

4.3.1. Position of the evaluator

Two positions are differentiated:

1. Evaluator and developer: evaluator and developer (of the object being evaluated) are rarely combined, except in glassbox methods. In competitions, ethics require these to be two different people.
2. Evaluator and user: (i) if the evaluator observes the user of the software in situation, evaluator and user are never the same person; (ii) the evaluator can temporarily adopt the position of user.

4.3.2. Evaluator expertise

Expertise is a complex notion, since one can be an expert in a particular domain (rarely in several), and even within a specific area there are variable degrees of expertise.

1. Non-expert evaluators are often used in methods with referentials, as they are given a set of points which need to be checked and then indicate the answers that match appropriately.
2. Expert evaluators usually intervene in methods with or without referentials, and they judge the quality/relevance of the answer in the given context.

In methods without referentials, expert evaluators will normally be required. However, it may be appropriate for the evaluators to be expert in evaluation but non-expert in the subject domain.

4.4. Conclusion

Our classification in § 3. is based on 3 questions (*What?, How?, Who?*). Each question consists of different sub-elements, for which distinct answers can be given. This may lead to a very high number of possibilities, if each combination of parameters is envisaged.

One could object that we have envisaged an exceedingly high number of procedures: (i) first this indicates the astounding abundance of the parameters which have been used in different frameworks; (ii) in actual fact, it is not the case, because some choices imply *de facto* other choices: the *glassbox* access is compatible only with experts as users; the evaluation of a practice is compatible only with *blackbox* access, etc. In the same way, the framework of a specific evaluation can significantly reduce the possibilities. If competitions are considered, some aspects are necessarily quasi-immutable: a competition which evaluates many entries is necessarily situated in the system approach (*cf. infra*), uses a blackbox method and applies quantitative measures.

These methodological distinctions allow a classification of approaches to be constructed. This classification actually has a hybrid status:

1. It is a tool which helps when revising or inventing evaluation procedures; one is obliged to stipulate major elements about which the evaluation procedure needs to formulate an opinion. This is what was experienced during our work on revising the EASA grid. It was fruitful for determining the nature of the objects to be evaluated, for eliminating spurious or ambiguous formulations and inappropriate criteria, and proposing new ones (David *et al*. 2005a, 2005b) for details on both the former and revised evaluation grids).
2. It is also an observatory of the knowledge constructed by the evaluation procedure: it indicates a way to apprehend and to reason about objects. It is particularly apparent when the consequences of different choices are explored and updated (*cf.* (ii) *supra*).

Table 1 is an exemplification of several approaches. It is not globally exhaustive: it does not show all of the possibilities, neither all of the currently existing ones, nor *a fortiori* the ones which do not (yet) exist. It is also not locally exhaustive, because it does not describe in detail the specificities of each procedure. But it clearly to shows two major things:

1. Evaluations always build points of view, which are always limited by the different chosen parameters. But choosing one or another parameter is justified by multiple reasons of differing natures (*cf.* § 2.). Consequently, every time an evaluation is conducted, it constructs its own object. The same spelling checker evaluated according to a developer procedure or in a competition will be observed in different ways. The chosen dimensions provide limited pieces of knowledge.
2. The objects to be evaluated only partially determine the evaluation which can be applied to them. The same spelling checker could be evaluated according to a developer procedure, or compete at TREC or EASA, or be evaluated according to practices and usage (that is why, in the table, we posit the same objects under all of the procedures).

Finally, the exemplification of the procedures, such as can be observed in Table 1, allows one to reflect upon the resemblances and differences between procedures. We shall now proceed with the general classification as such.

5. General classification: system vs. usage[2]

Two major approaches can be identified: system and usage.

1. In the *system* approach (white background in Table 1), the intrinsic performance of the software prevails; evaluation of the usage within a real context (professional, private, collective, etc.) is excluded, the user is not taken into account, nor is the diversity of the users (employees, students, etc.) or the usages (occasional, regular, etc.). This does not imply that aspects which concern users directly are not covered (interface ergonomics, installation, etc.), but that they are fairly limited and, if the user is indeed considered, it is always from the standpoint of a *potential* user. In this approach, one focuses on an ideal/norm where each object is posited at a certain distance from this ideal/norm. Objects can then be compared (when there is only one object, the comparison is of course lost). The norm could be represented by referentials or qualitative judgments. The objects to be evaluated are reduced to aspects that are measurable, comparable, and that generally belong to one field. Only very few dimensions are considered, so evaluation procedures often "abolish" the complexity of objects. All evaluations conducted in competitions use the system approach.
2. In the *usage* approach (grey background in Table 1), thorough preliminary meditation on the "objects to be evaluated" is crucial. The item of software itself may not be directly considered, but more general practices surrounding the usage of the software are addressed (the question marks after the name of the systems in Table 1 refer to this). One then focuses on the complexity of the situation (including the object): the multidisciplinary aspects, the specific tasks aimed at specified users, the interactive properties, etc. In this case, objects are considered as practical complex devices, *i.e.*, a complex set of social and technical relationships, which are established between groups or individuals and technical objects, including representations, norms, and habits (Amar, 2000; Le Marec, 2001). This approach can be used when questions related to user practice within a given context are addressed (*e.g.* museums, educational situations in which the pedagogical and relational approach is also studied, etc.). The perspective here is radically different, compared to the system approach. It is a different type of knowledge which is exhibited.

[2]We prefer the term *usage* to that of *user*: the former implies the latter, and puts more emphasis on social practices rather than on individual or cognitive characteristics.

To illustrate these two types of knowledge, one can think of the spelling checker in Microsoft Word™. Everyone has experienced its shortcomings. In a developer approach or in a competition, one could exhibit them precisely, and perhaps be tempted to assign a negative judgement. On the other hand, in a usage approach, one could exhibit its utility and its context of use, also including the reasons why it is used in spite of its defects. One perceives with this example how different knowledge is objectified and how difficulties are encountered when choosing an approach, precisely because specific points of view are constructed: either the tool is "invalidated" for (very) good reasons, even if it is the most widely used globally; or it is "validated" despite its faults. In both cases, the objectified knowledge is situated within two radically different perspectives.

6. General conclusion

The outlines indicated may make a helpful addition to general classification techniques in relation to evaluation procedures. As it is a classification which can be defined as methodological, comparisons become possible, and the most common evaluation approaches can then be analysed. We have shown that evaluation always constructs a point of view: because this point of view is limited (it chooses some dimensions, but never all of them) and because the objects to be evaluated are complex, the latter can be submitted to different approaches. In this sense, any evaluation always constructs its own object, and the objects to be evaluated only partially determine the evaluation which can be applied to them.

We also address the issue of the epistemological nature of evaluation. If one agrees that evaluation is a technique, and that it may become the subject of applied research, what can one conclude? Three attitudes seem feasible: consider evaluation to be an engineering science, or just a plain science, or a methodological branch of a science. In this paper, we have chosen to explore the third attitude. We have clarified some of the problems, but further in-depth research is necessary in order to specify more precisely the epistemological status of evaluation.

7. Acknowledgements

The following institutions have sponsored this work; CNRS, EKMA, The Higher Education Academy, The Joint Information Systems Committee, Lancaster University, Universités Paris 10 & Montpellier 3. We would especially like to thank the three anonymous referees and Jean-Luc Minel, whose questions helped us rethink certain aspects of our work.

8. References

Abbou A. (2000), « Evaluation des résumeurs automatiques disponibles », *La Tribune des industries de la langue et du multimédia*, 35-36, 2-7.

Adda G., Lecompte J., Mariani J., Paroubek P., Rajman M. (2000) « Les procédures de mesure automatique de l'action GRACE pour l'évaluation des assignateurs de partie du discours pour le français », in Chibout K. Mariani J. Masson N., Néel F. (éds) *Ressources et évaluation en ingénierie de la langue*, Paris : Duculot, p. 645-664.

Amar, M. (2000), *Les fondements théoriques de l'indexation : une approche linguistique*. Paris: Editions de l'ADBS.

Amar, M., David S., (2001), *Evaluation de logiciels d'extraction dans les champs de l'indexation, la traduction et la terminologie. Corpus INRA*. Rapport de recherche. Action de recherche concertée (n° X/7.10.04/llec.A3o), AUF & CNRS UMR 8529 (Cersates, Université Lille 3), 109p.

Bar-Hillel Y. (1960). "The Present Status of Automatic Translation of Language", *Advances in Computers*, Vol. 1, New York Academic Press, 91-141.

Chaudiron S. 1999. « Réflexions préalables à l'analyse qualité des logiciels d'ingénierie linguistique ». *Bulag*, 24, 153-168.

Chaudiron, S. (Ed.) (2004), *Évaluation des systèmes de traitement de l'information*, Paris : Hermès.

Chaudiron S., Mustafa el Hadi W. (2007), « L'évaluation des outils d'acquisition de ressources terminologiques : problèmes et enjeux », in *Actes de la première conférence TOTH*, Annecy 2007, 163-179, `http://www.porphyre.org/toth/07/actes`.

Cori, M., David S., Léon J. (2002). « Pourquoi un travail épistémologique sur le TAL », TAL, *Problèmes épistémologiques*, Cori M., David S., Léon J. (eds), 43 (3), 7-22.

Cori M., Léon J. (2002), « La constitution du TAL. Étude historique des dénominations et des concepts », *Problèmes épistémologiques*, 43 (3), 21-55.

David, S., Panckhurst, R. (2004), "Comments on the current EASA evaluation process", Talk, European workshop, Montpellier, France, November 2004, *Evaluation in e-learning: review & future directions*, `http://www.univ-montp3.fr/~rachel/spip/article.php3?id_article=3`

David, S., Panckhurst, R., Whistlecroft, L. (2005a), "Many Forms of the Future. A report on future options for the organisation of EASA", Report submitted to EKMA, Oxford, April 11 2005, 45p.

David, S., Panckhurst, R., Whistlecroft, L. (2005b), "Revising the Evaluation Procedure of the European Academic Software Award", European University Information Systems, Proceedings, EUNIS 2005 Conference, 20-24 June 2005, The University of Manchester, `http://www.mc.manchester.ac.uk/eunis2005/medialibrary/papers/paper_111.pdf`

EASA : `http://www.easa-award.net` (This official weblink is no longer valid. The most recent EASA competition website can be viewed at: `http://www.bth.se/llab/easa.nsf`).

Ellis D. (1992), « The Physical and Cognitive Paradigm in Information Retrieval Research », *Journal of documentation*, 48 (1), 45-64.

Falkedal, K. (1998), "Evaluation Problems from a Developer's Point of View", in Nübel, R., Seewald-Heeg U. (eds), *Evaluation of the Linguistic Performance of Machine Translation Systems*. St-Augustin: Gardez! Verlag, 137-150.

Fondin, H. (2006), « La science de l'information

ou le poids de l'histoire », article inédit diffusé le 24 mars 2006, disponible en ligne : http://w3.u-grenoble3.fr/les_enjeux/2005/Fondin/index.php (consulté le 18 mars 2008).

Habermas J. (1973) (1968), *La technique et la science comme "idéologie"*, Paris : Gallimard.

King M., Falkedal K. (1990), « Using Test Suites in Evaluation of Machine Translation Systems », *Coling*, vol. 2, 211-216.

Language and Machines. Computers in translation and linguistics (1966), A report by the Automatic Language Processing Advisory Committee (ALPAC), National Academy of Sciences, National Research Council.

Lavenus K., Lapalme G. (2002), « Evaluation des systèmes de question réponse. Aspects méthodologiques ». *TAL : Problèmes épistémologiques*, 43 (3), 181-208.

Le Marec, J. (2001), « L'usage et ses modèles: quelques réflexions méthodologiques », *Spirale*, 28, 105-122.

Le Marec, J. (2004), « Les études d'usage », in Chaudiron S. (ed.), *Evaluation des systèmes de traitement de l'information*, Paris : Hermès, 353-372.

Medida Prix: http://www.medidaprix.org

Milner J.-Cl. (1989), *Introduction à une science du langage*, Paris : Le Seuil.

Minel J.-L. (2004), « Évaluation des systèmes de résumé automatique », in Chaudiron S. (éd.), *Évaluation des systèmes de traitement de l'information*, Paris : Hermès, p. 171-184.

MUC: http://www-nlpir.nist.gov/related_projects/muc/

Nübel R., Seewald-Heeg U. (éds) (1998), *Evaluation of the Linguistic Performance of Machine Translation Systems*, St-Augustin : Gardez ! Verlag.

Panckhurst, R., David, S., Whistlecroft, L. (Eds), 2004, *Evaluation in e-learning: the European Academic Software Award*, Montpellier : Publications de l'université Paul-Valéry, http://www.pulm.fr/evaluation-in-e-learning-easa

Pariente J.-CL. (1973), *Le langage et l'individuel*, Paris : Armand Colin.

Sparck-Jones K., Gallier J. R. (1996), *Evaluating Natural Language Processing Systems: an Analysis and Review*, Berlin: Springer-Verlag.

TREC: http://trec.nist.gov/ (especially: http://trec.nist.gov/pubs/trec15/t15_proceedings.html:Voorhees E. M., Overview of TREC 2006).

Appendix

Table 1. Exemplification of some approaches.

Approaches		developer	TREC	EASA	usage
		↓	↓	↓	↓
What	*Evaluation object*	one item of software	several items of the same type of software	several items of different types of software	practice
	Access	glassbox	blackbox	blackbox	for the software: blackbox
How	*Object distribution*	individual	individual	individual	for the software: individual
	Resources	with referentials	with referentials	without referentials	for the software: without referentials
	Measures	quantitative measures (true/false answers)	quantitative measures (true/false answers)	quantitative measures (grid)	surveys
	Evaluation distribution	single	single	aggregated (stage 2) and collective (stage 3, finals)	collective
Who	*Evaluator position*	evaluator ≠ user	evaluator ≠ user	evaluator ≠ user but temporarily so (stage 2)	evaluator ≠ user
		evaluator = developer	evaluator ≠ developer	evaluator ≠ developer	evaluator ≠ developer
	Expertise	experts	non experts	experts (stages 2 & 3) and non experts (stage 3, finals)	experts
		↓	↓	↓	↓
Type of software (which could be) evaluated		all software	spelling checkers QA* systems MT** systems	spelling checkers QA systems MT systems	spelling checkers? QA systems? MT systems? interactive information points (museums)?

*QA: question/answering; **MT: machine translation; ?: the software may not be primary focus of evaluation; grey background = usage approach; white background = system approach.