Stimulating Collaborative Development in Operations Research with libOR

Kris Ven, Kenneth Sörensen, Jan Verelst
University of Antwerp
Antwerp, Belgium
{kris.ven, kenneth.sorensen, jan.verelst}@ua.ac.be

Marc Sevaux
University of Valenciennes
Valenciennes, France
marc.sevaux@univ-valenciennes.fr

Abstract – In this paper we describe the development of libOR, an on-line library for the operations research (OR) community. The design and operation of this website is inspired by the Open Source movement and recent developments such as Wikipedia. In operations research, data sets are exchanged between researchers in order to test the performance of newly developed algorithms. Currently, the exchange of these data sets suffers from many problems. One of the main problems is that data sets are currently exchanged through a centrally maintained website, which makes it slow to respond to new developments. By applying an Open Source approach to content creation, we hope to spur the diffusion of information within the operations research community.

I. INTRODUCTION

The Open Source Software (OSS) movement introduced a new way to develop software. Although many people were initially rather skeptical towards it, much research has documented the inner workings of OSS communities and their practices (see e.g. [1,2]). Nowadays, the Open Source development model has proved to be a viable alternative to traditional software development, especially in distributed environments. Some authors have described how traditional software engineering can benefit from using OSS tools and practices [3]. Hewlett-Packard for example has indeed started using some of these practices — albeit in a modified form — for their internal development and in their collaboration with partners [4]. Some have reported advantages of using an OSS-like model such as a decrease in the number of bugs and increased learning from other project members by sharing practical coding tips [5].

In the last 3–4 years, the Open Source idea has been extended to support creative work in domains other than software development. The initiatives of Wikimedia1 and Creative Commons2 are well-known examples. The total number of articles in Wikipedia (342,000 articles as of October 2004) clearly shows the potential of this approach.

Some authors (see e.g. [6,7]) have compared the OSS movement with academic research. In both cases, the primary motivation of the participants is not a financial reward. Moreover, researchers around a certain topic tend to form a virtual community in order to stimulate further research. This is very similar to what happens in Open Source Software development. Finally, in both cases, participants gain respect in the community by publicly releasing the result of their activities to the rest of the community. In research, this is an important step to validate scientific results. This is once again quite similar to the OSS community, where each developer deliberately releases his/her code to the community, so other members of the community can further build upon this code and improve the overall quality of the software.

Some authors have noted that thanks to the similarity between academic software research and Open Source Software development, academic software research may benefit from adopting certain Open Source practices. These benefits can be obtained by using OSS tools, assuring the quality by performing peer reviews and by following a community-driven development model [8]. In our opinion, certain other (non-software) domains of scientific research as well can benefit from utilizing Open Source practices. In many research domains, information is stored distributed and heterogeneously on the Internet. However, in order to achieve economies of scale, this information could better be integrated and further developed by the collaboration of the entire research community. The Open Source development model provides the necessary tools and practices that can be used in an academic (non-software) environment. In operations research, researchers exchange data sets corresponding to optimization problems. These data sets are used to test and compare the results of new and existing optimization algorithms. We have implemented an on-line library for the exchange of such data sets that offers many advantages and operates in a way closely related to that of a typical Open Source community.

The rest of the paper is structured as follows. We will start by describing the operations research domain and the problems researchers in this domain are facing. Next, we will present libOR, which is a virtual library for operations research, based on an Open Source collaborative model. We will present the organizational as well as the technical construction of the website, and our experiences with libOR so far. Finally, conclusions are drawn for the future challenges of libOR.

II. OPERATIONS RESEARCH AND OPTIMIZATION

Operations research (OR) can be defined as the application of scientific methods to aid in the process of decision making. OR originated in the second world war, when scientists realized that the challenges posed by the enormous logistical operations (hence the name operations research) could be at least partially solved by applying mathematical methods. After the war, similar methods were developed and applied to problems in business and industry.

Optimization constitutes a large part of the broader discipline of OR (other parts include simulation, statistics, graph theory and queuing). In optimization, algorithms and heuristics are developed to find a solution which maximizes or minimizes the value of an objective function subject to some constraints. The difference between a heuristic and an (exact) algorithm is that the former does not offer the guarantee that the solution it finds is optimal (i.e. the best possible solution). A famous example of an optimization problem is the traveling salesman problem (TSP). In this problem, a set of spatially distributed “cities” are given and the distance between each pair of cities is assumed to be known. The objective of this problem is — starting from a

1 http://wikimediafoundation.org/wiki/Our_projects
2 http://creativecommons.org

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), pp. 71-75
given city — to find the order in which to visit all cities and return to the starting point in such a way that the total distance traveled is minimized. Although the problem is very easy to state and has many practical purposes in such diverse areas as logistics and genome sequencing, it is very hard to solve in that the computation time of all known algorithms rises exponentially with the size of the problem. Recently, state-of-the-art methods succeeded in finding the optimal solution to a TSP consisting of all 24,978 cities of Sweden. The computation required to find this solution was performed on a large cluster of parallel computers, but would have required almost 85 years on a single Intel Xeon 2.8 GHz.\(^2\)

The excessive computational time required to find the optimal solution to a large number of interesting problems has given impetus to the development of heuristics, that attempt to quickly identify a solution that is "good enough." More recently, the development of more advanced heuristics, commonly known as metaheuristics, has received increasing attention\(^3\).

To test the performance of newly developed algorithms and (meta)heuristics for a specific problem, they are generally applied to a suite of test problems. The importance of having high-quality test problems readily available cannot be underestimated. There is some truth to the often-heard complaint that a paper in optimization is only considered publishable if the heuristic developed in it succeeds in finding at least one solution that is better than the best solution previously known. The current way in which these data sets are distributed however, leaves much room for improvement.

Test problems are commonly distributed as plain text files, an example of which can be seen in Fig. 1. The fact that such a file lacks all semantic information about what each number represents, may cause several problems. Input routines generally have to be rewritten for every different suite of test problems, as the syntax is often slightly different. Moreover, the lack of semantic information is the source of many programming errors that are very difficult to trace. An example of such an error is reading the wrong numbers into the wrong variables (e.g. reading travel times as customer demand and vice versa in a TSP). The possible occurrence of such errors leads to a long and arduous debugging cycle. More generally, the fact that there is no way to identify a file as a valid instance of a data set for a specific problem instance, makes it difficult to judge whether an observed problem should be traced to the algorithm or to the data set.

A second problem with the current way of exchanging problem data sets is the fact that test problem suites are scattered across the Internet. The closest to a central repository of problem data sets is the so-called TSPLIB\(^4\) which was first described in [9]. This database is maintained by a single person, making it very slow to respond to current developments. Moreover, it is far from being a central library for all data sets. Data sets for the TSP for example, are not included and are distributed through the so-called TSPLIB\(^5\). Also, it does not contain any results obtained by different heuristics on different problem suites.

In order to tackle these problems, we started the libOR initiative\(^6\). The primary aim of this project was to stimulate information sharing within the OR community. There were technical as well as organizational problems that needed to be solved. In this section we will focus on the technical problems.

The main issue that needed to be resolved was the format of the data sets. To improve the interchangeability of the data sets, we chose to save the data sets in XML format (see Fig. 2). This ensures that files can be easily exchanged between different computer platforms. More importantly, the validity of XML files can be checked against an XML Schema Definition (XSD). For each type of problem, an XSD is constructed. As we will see later, researchers can submit new data sets for a certain problem. When a new data set is received, its validity is immediately checked against the XML Schema Definition. Only data sets that pass this test are accepted on libOR. Another advantage of using XML is that the data sets contain semantic information, which makes the data sets easier to read, and possible errors in the file can be more quickly identified and corrected.

The switch from using plain text files to XML files means that the data sets cannot be read and processed directly, but must first be parsed. This will require a change for most researchers. However, there are many excellent Open Source XML parsers (such as Xerces\(^7\)) available for different programming languages. This should facilitate the transition for most researchers.

The library itself is organized around a tree structure. A simplified entity-relationship model of this structure can be found in Fig. 3. At the highest level, we distinguish between different problem categories. Each category refers to a set of similar problems, e.g. routing problems. Each category contains several problem definitions. A problem definition is a well-defined type of problem, assigned to one or more problem categories. Each problem definition is assigned an XML Schema Definition. An example of a problem definition is the capacitated vehicle routing problem with time windows. For each problem definition, multiple instance collections can be submitted. An instance collection is a grouping of a set of similar instances. The term instance refers to single data set in XML format, containing test data for a specific situation. All data sets (XML files) of an instance collection must conform to the XML Schema Definition defined in the problem definition. The data sets contained in an instance collection will be used by OR researchers to test the performance of different algorithms and heuristics. The results obtained by OR researchers can be submitted as a result collections. This makes it easy to compare the performance of different algorithms and heuristics on the same data set.

### III. LIBOR

\[^2\]http://www.tsp.gatech.edu/sweden
\[^3\]http://www.brunel.ac.uk/depts/ma/research/jeb/info.html
\[^4\]http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
\[^5\]http://webhost.ua.ac.be/libor/
\[^6\]http://xml.apache.org/

Fig. 1. A typical data set for a "Steiner tree problem in a graph"

---

\[^1\]http://www.tsp.gatech.edu/sweden
\[^2\]http://webhost.ua.ac.be/eme/
\[^3\]http://www.brunel.ac.uk/depts/ma/research/jeb/info.html
\[^4\]http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

Proceedings of the First International Conference on Open Source Systems
Genova, 11th-15th July 2005
Marco Scotto and Giancarlo Succi (Eds.), pp. 71-75
IV. CONTENT UPDATING

The other problem concerns the organization of the OR library. It is an impossible task for a single person to maintain and update a complete library of OR problems. To enable libOR to keep up with new developments and to contain data sets for the complete OR domain, we choose for decentralized updating of the website. The main objective of the project was to provide a platform through which all members of the OR community can collaborate and build a library of OR test problems. Initiatives such as Wikipedia have illustrated that a decentralized model in which updates are performed by a large number of volunteers can indeed succeed.

In libOR, we used an open community model, similar to the approach used by Wikipedia: every researcher within the OR community can submit new content to the libOR website and help in the further development of the library. In scientific research, the validity and correctness of the data offered is of primary importance. Hence, the data needed to be protected from possible misuse. In addition, some control about the future expansion of libOR was preferred, to avoid a proliferation of categories and problems. This means that some additional workflow model was required to control the processing of information. A completely centralized approach would contradict too much the otherwise open character of libOR. Therefore, it was decided to assign a responsible for each problem category. The responsible is in charge of reviewing new content that is submitted to his/her section of the website. Every member in the OR community can submit new data sets for existing problems, submit results of an algorithm he/she obtained for a certain data set and propose new problem definitions. Once a new data set or new problem definition is submitted, it assumes a “pending” state (see Fig. 4). Each responsible can retrieve a list of pending data sets and problem definitions and review their contents. Only verified and approved items will be accepted and displayed on the website. Rejected items will be hidden from the website.

The decision on whether to accept or reject a new problem definition is not made by the responsible himself. Once a new problem definition is submitted, a thread on a discussion forum will be opened. Every member in the OR community can discuss the current contents of the problem definition. This discussion concerns the exact definition of the problem as well as the structure of the XML Schema Definition. Once the community has given its consent, the problem definition is accepted.

The process described here is based on the peer review process that is widely used in Open Source projects. The

V. IMPLEMENTATION

From the start of the implementation of libOR, we decided to use Open Source Software (OSS) where possible. Many reliable ready-to-use Open Source packages exist, which can speed up the development process considerably. The Open Source licenses also provide the possibility of adapting the software to meet certain specific requirements. A final reason for using Open Source Software is financial. Using Open Source Software to build libOR meant that development and operating costs could be kept to a minimum. This, together with the hosting support by the University of Antwerp, allows us to distribute libOR and all the data it contains free of charge.

The first decision we took was the platform on which the website would run. We chose a basic LAMP (Linux, Apache, MySQL, PHP) configuration for two reasons. First, we already had positive experiences with similar configurations in the past. Second, PHP is a relatively easy-to-learn language, which makes it easier to attract new developers.

During the development, we made use of three more
OSS packages. As already mentioned, we use Xerces to verify the validity of uploaded XML data sets. When a parsing error occurs, the user is informed about this, and is given the output of the parsing process. This should help the researcher in improving his/her data set.

Since OR makes extensive use of mathematical formulas, it makes sense to support their inclusion in certain text fields on libOR. Since displaying mathematical formulas in most browsers that are used nowadays is still problematic, even when using MathML, we had to look for another solution. We therefore used the texvc program, which is part of the Mediawiki\textsuperscript{10} project. This program supports rendering LaTeX formulas to representations that are safe to use in today’s browsers. Currently, we use it to translate LaTeX formulas in PNG (Portable Network Graphics) images.

As already mentioned, libOR also features a discussion forum on which – amongst others – propositions for new problem definitions are discussed. Many Open Source discussion forums written in PHP are currently available. Our main concern was to have a relatively simple forum, without too many features or too much administration that could be easily integrated into the rest of the website. Phorum\textsuperscript{11} was selected as it fulfills the requirements stated before.

Early in the development we decided to release the source code of libOR under an Open Source license. The code base for libOR was given the name “ORLi” and released under the GNU GPL. Our main motivation for releasing ORLi as Open Source Software was to emphasize ORLi as Open Source Software was to emphasize with everyone, we are inviting each member in the community to take part in the further development of libOR. We hope this will speed up the development process and that each member in the community can propose and implement new features, without having to rely on a central administration. An often claimed advantage of Open Source Software is Linux’s law which states that “Given enough eyeballs, all bugs are shallow.” [13]. If enough users are motivated to report bugs encountered with libOR, the quality of the application will increase. Given the limited programming background of the potential users, we do not expect them to get very much involved in the bug fixing process itself. On the other hand, another Open Source initiative within the OR community exists. The COIN-OR project\textsuperscript{12} aims to provide an environment in which implementations of optimization algorithms are released under an Open Source license. This facilitates building on previous research and to improve the currently used algorithms without having to re-implement them. The COIN-OR project suggests that it is possible for OR researchers to become active in the field of software development.

In order to support this community-driven development model, we will provide a bug tracker and access to the Subversion repository in the near future.

VI. EXPERIENCES

The libOR initiative was first presented at the Workshop on Real-Life Applications of Metaheuristics in December 2003 [14]. In October 2004, it was presented at the INFORMS annual meeting in Denver, the largest conference on operations research [15]. At this moment, there is no production version of libOR, as we are still in the process of finding enough responsibilities for each problem category, but the “sandbox” version\textsuperscript{13} is quite active. Although considerable interest has been raised by this project, it is not easy to find people who are willing to invest time in libOR and manage a problem category. The most important reason for this is probably the fact that the knowledge of XML possessed by the average operations researcher is rather limited. The fear of having to convert all plain-text data files in a category to XML may seem a daunting task. Creating the corresponding XML Schema Definitions probably poses an even greater challenge. Also, there seems to be an unjustified concern that writing programs that work with XML input files is much more difficult than writing programs that read the corresponding plain text files.

We are confident however, that in time the knowledge of XML will increase and that the use of plain text files for the interchange of problem data in optimization will fade away. The use of an external parser to read data files in XML will require a learning effort from the optimization community, but we are convinced that the benefits (especially the validation of data files) greatly outweigh the drawbacks.

In terms of scientific advantages, we expect libOR to facilitate co-operation among operations researchers and increase rapid proliferation of scientific advances throughout the community. A researcher wishing to create an algorithm or a heuristic for a problem he/she has not worked on before now faces the tremendous task of searching the literature for every paper in which this problem is tackled. libOR, in its capacity to become the “one-stop” repository for all attempts to tackle any problem, may dramatically improve upon this situation. Another potential advantage of libOR is the possibility to “publish” the results of unsuccessful attempts to tackle a problem. As mentioned, journals in optimization tend to publish only papers that present methods that improve upon the best-known methods. The result of this is that failed attempts to apply a specific method to a specific problem are never released. This in turn leads to the inefficient situation that every researcher goes through a frustrating “learning period” in which he/she makes the same mistakes many have made before. The possibility to report the results of failed attempts may reduce the length of this learning period.

\textsuperscript{10} http://wikipedia.sourceforge.net
\textsuperscript{11} http://www.phorum.org
\textsuperscript{12} http://www.coin-or.org
\textsuperscript{13} A non-production version of libOR in which users can test and get familiar with the functionality of the library.
VII. CONCLUSIONS

In this paper we have presented libOR, an on-line library for the operations research community, which is related in two ways to the Open Source development model. First, during the implementation, we made exclusive use of Open Source Software. The implementation framework itself (ORLi) is also released under the GPL license. Second, the content that will fill the library is created in a collaborative effort by the OR community, and will be subjected to a peer review process in which each member of the community can take part.

The success of libOR will ultimately depend on the collaboration of the OR community. We have experienced certain difficulties in attracting the initial responsibilities for the problem categories, but interest seems to be on the rise and we expect to gain momentum in the near future. In this environment, it is important to reach a critical mass of volunteers that initiate the first content. By doing so, the advantages of the approach will become more tangible for other researchers who will become more likely to join the community. However, we foresee that it will remain difficult to attract new volunteers, since researchers are geographically distributed, and direct contact with other members of the community is rather difficult.

An interesting, open question for the future is whether members of the OR community will become actively involved in the further development and expansion of the source code of libOR (ORLi). Since the research domain is not immediately IT-related, it seems unlikely that many patches will be submitted. However, the COIN-OR initiative seems to illustrate that this model could succeed in operations research. In either case, it remains interesting to see whether and which Open Source practices can be used to leverage research in the OR community or other domains of academic research.

VIII. REFERENCES