ADEPT

Analysis and Development of Electronic Publishing Technologies Project Unit for Scientific Information and Learning, KTH, Stockholm, and Department of Computer and Information Science, Linköping University

Erik Sandewall

Domain Modelling of Rights and Conditions for Parallel Publication of Research Articles

Final Report from Phase I of the oa.se Project

This series contains technical reports from the joint KTH/Linköping project Analysis and Development of Electronic Publishing Technologies (ADEPT). The present report can persistently be accessed as follows:

Memo persistent URL: Date of manuscript: http://piex.publ.kth.se/reports/adept/005/ 2010-02-19

Related information can be obtained through the following www sites:

KRF website:http://piex.publ.kth.se/krf/The author:http://www.ida.liu.se/~erisa/

1 The Workflow around the Journal Information Resource

The project "Domänmodellering av rättigheter och bivillkor vid parallellpublicering av vetenskapliga artiklar" (Domain modelling of rights and conditions for parallel publishing of scientific articles) has been approved for two phases, the first phase ending by the end of year 2009, and the second phase starting at the beginning of 2010. The present is the main report from the project's first phase.

Figure 1 (below) shows how the Journal Information Resource (JIR) that is the intended result of the project, is supposed to interact with related systems, according to the design at the end of Phase I. This structure will be described in Section 1 of this report. Section 2 addresses the acquisition and maintenance of the system's knowledgebase; Section 3 is a summary of the project history during Phase I, and Section 4 describes briefly the system demonstration that was done in February, 2009 as well as the software testbench that was used for the demo.



1.1 Intended Workflow around the JIR

The purpose of the Journal Information Resource (JIR) is to provide institutional repositories and other similar systems with information that they need in order to correctly respect publisher's parallel publishing conditions when they pre-publish or post-publish research articles whose primary publication is in a scientific journal. The intended workflow as seen from the JIR point of view is as follows. The Repository acquires the full text of an article together with its bibliographic and other relevant metadata, including in particular an identifier for the journal (journal name, or ISSN number), year and month of publication, and IPR-related aspects of how the research in the article was funded. The Repository sends an information request to the JIR including relevant parts of the metadata. The JIR returns information to the Repository, indicating whether parallel publication is consistent with its knowledge about the rules of the publisher of the journal in question, and what particular measures the Repository must attend to in order to be compliant.

Many publishers impose constraints as to which type of repository is allowed to parallel publish an article, restricting it to e.g. non-profit repositories, or repositories of an author's employer. The JIR can take these constraints into account based on its knowledge of the character of each client Repository, and using information in the information request concerning the employment status of authors.

Many publishers also require a certain so-called embargo period between the date of publication in the journal and the date of parallel publication. The JIR takes this into account using the provided bibliographic metadata.

A number of publishers impose particular restrictions on which version of an article may be used for parallel publication, and which graphical appearance is allowed. The versions of the article include the one that was first submitted for publication (pre-reviewing), the one that resulted after one or more rounds of reviewing, and the one that resulted from the copy-editing done by the publisher. Publishers may impose constraints on which of these may or may not be used for parallel publication.

The graphical appearance may also be restricted. Some publishers explicitly require that the exact graphical version that is used on the publisher's server shall also be used in the Repository, whereas some others explicitly prohibit this and require that another graphical appearance be used. In general, information about version and appearance is provided in the return message from the JIR system, based on the contents of its knowledgebase.

Finally, a certain number of publishers require that the posted version of an article in a repository must provide certain information on its first page. This may include the basic bibliographic information, or mention of the original publisher as the copyright holder and of how it can be reached. Furthermore, some of them require that not only shall certain information be given, but furthermore that a certain standardized phrase must be used literally. These phrases are sometimes as long as ten lines, and they may require the bibliographic information to be inserted inside them.

In order to accomodate such requirements, the JIR maintains a register of *front-page scripts*, each of which specifies the wording and the rules for inserting metadata that are required by a particular publisher.

There are several possibilities with respect to the division of tasks between the JIR and the Repository. One possibility is to let the Repository maintain a copy of the register of frontpage scripts, so that the JIR only needs to transmit an identifier for the choice of script when one is required. This is the case that is illustrated in the figure on the previous page. Another possibility is to let the JIR transmit the full script to the Repository each time one is needed, and a third possibility is to let the Repository transmit full bibliographic information to the JIR, including names of authors and title of the article, and to let the JIR instantiate the script, format the front page of the article, and return the ready-formatted page in pdf form.

We believe that the third alternative is unlikely since probably each repository will prefer to have its own frontpage generator and frontpage design anyway. The choice between the first two alternatives is a matter of convenience. The interposition of a separate 'labmop' entity, which will be discussed in Section 1.3, will add a further possibility to this choice.

1.2 Obtaining the Information about Rights and Conditions

In order to automate the described workflow, it is necessary for the JIR to have access to the information about rights and conditions in *structured form*, so that it can be used by the JIR software. In principle there are three sources of this information:

- Information provided by the author(s) of the article, who have signed a contract with the publisher
- Information provided by the publisher, either on their website, or as an answer to information requests from the operators of the JIR system
- Information provided by the Romeo-Sherpa database and website

We are not aware of any other service besides Romeo-Sherpa that can be used for this purpose. For the present project we disregard the first option, and focus on the other two.

The Romeo-Sherpa database and website is very widely used for the workflow described here, when it is done manually. There is no trivial way of automating it, since much of the information is provided in English-language phrases and not as a formal structure, and since some of the information is only provided with a phrase of the type "for details, see the publisher's website." This applies in particular for frontpage scripts.

The Romeo-Sherpa database is quite large, but it does not cover all publishers which is natural in view of the large number of scientific journal publishers in existence. More significantly, it only provides publisher-level information, which is a problem since one can not assume that all journals of a certain publisher apply the same rules; the opposite case appears not to be uncommon.

In view of this, our project has used a two-pronged approach. On one hand, we have built a knowledgebase containing formal encodings of parallel publishing conditions, including references to frontpage scripts, for use by the JIR software. The information in this knowledgebase has initially been obtained from a downloaded copy of the Romeo/Sherpa database, using a combination of automatic and manual steps. It has been extended with additional information obtained from the websites of some of the publishers.

At the same time, we have also built a software tool that accesses the Romeo/Sherpa database dynamically in order to obtain the information for a particular publisher in the form of English-language phrases, and which decodes these phrases so as to extract the desired information.

Both these methods are used in the present system, and they complement each other. Using the JIR knowledgebase has the advantage that it provides more precise information, and that it can be gradually extended with more information besides what is provided by Romeo/Sherpa, for example journal-specific information, or information that has been negotiated separately by our users in some situation. The direct, dynamic access to the Romeo/Sherpa database misses some information, but it has the advantage of always being up-to-date with the latest amendments of Romeo/Sherpa.

1.3 The Author's Perspective on Workflow

In order to have a successful system for parallel publishing, as well as other aspects of open access, it is important to take the perspectives of the authors into consideration, and not merely the perspective of the university library as an operator of the institutional repository. This observation is particularly relevant in the present context, since our task is complicated by the issue of versions and graphical appearance of articles, and since the repository perspective is more narrow than the author perspective in this respect.

During the sequence of steps that lead up to a finished article, the author must administrate a number of versions of the article, and this number may sometimes be fairly large. There may be early versions of the article that are circulated to a few colleagues for comments before the article is submitted for publication. (There is a difference between disciplines in this respect: this practice is common in some disciplines and is said to be almost unthinkable in others). Some journals may require the submitted version of an article to be formatted in a particular way, in particular if the journal applies double-blind reviewing. If the author disseminates a preprint in the sense of a pre-reviewing version of the article, then she may wish the preprint to contain the URL for a designated webpage where the forthcoming bibliographic information for the accepted article can be posted, so that the reader of the preprint can look it up and make a correct citation when the time comes. This list can easily be extended.

The on-line version of this report (see URL in the front page) contains links to sample frontpages for a number of different versions of a sample article.

These considerations lead us to propose that there is a need for an organizational entity that assists authors with the management of successive versions of their articles, including the proper formatting of the articles, the controlled circulation of them, the posting of them on local websites, submission to a journal and keeping track of the contractual aspects, and so forth. We use the term *Laboratory-level Management of Publications*, or LABMOP, for this entity, recognizing that a "laboratory" at least in Swedish parlance can be anything from a research group of half a dozen people, to an entity with several hundred employees. In this way we avoid any commitment as to the size of group that is supported by the LABMOP.

If such a LABMOP is organized, then it seems appropriate to let it be the entity that applies the conditions for parallel publication to each specific article, which means that it becomes the client and user of the Journal Information Resource. The relationship between the LABMOP and the Institutional Repository in the university in question may be conceived in a variety of ways. One may set up the Repository so that it in fact performs the LABMOP services as well. One may also have separate entities with a trustful relation between them, so that the Repository receives checked-out and ready-formatted documents from the the LABMOP, and where the Repository is responsible for posting the article, reporting to national and other global databases, and long-term preservation.

The diagram on page 1 shows one single box for the Institutional Repository and the LABMOP, and leaves the details of their relationship to be worked out in each case.

We have seen the need for an entity such as the LABMOP in earlier projects and in several contexts, so this development is not specific to the present project. On the other hand, because of the variety and character of the publishers' parallel publishing rules, and in particular the need for frontpage scripts, we feel that the present project would have been unduly restricted if we had not introduced the possibility of a LABMOP into its conceptual structure as well.

We assume a relatively large set of services in the entity labelled "Institutional repository or labmop" in the diagram above. If it is felt that implementing these services in the Institutional Repository would take too far, then either a LABMOP-type entity will be required, or authors must be instructed to perform these tasks themselves.

In summary, although the initial plan for this project was to develop a Journal Information Resource for use by institutional repositories, in the course of work we have arrived at the conclusion that in some contexts it may be appropriate, or even necessary, to introduce a separate type of organizational entity which is intermediate between the authors and the repository, and which is here called the LABMOP.

1.4 Documentation of the Journal Information Resource

We have created a composite webpage for current information about the Journal Information Resource, located at

```
http://www.ida.liu.se/ext/jir/
```

This persistent webpage contains general information about the JIR software, definitions of interfaces, documentation for the software and the knowledgebase, and so forth. It contains a link to the currently operating version of the JIR server, which is at present

http://piex.publ.kth.se:18086/

The persistent webpage also contains a link to the currently operating version of the facility that dynamically interprets the contents of the Romeo-Sherpa database for a particular publisher. This facility is presently a cgi-bin service located at

http://www.sics.se/

The JIR composite webpage is organized so as to be persistent, in line with the general policy of our Experimental Electronic Press. The addresses of the servers are not persistent and are likely to change in the future, but the JIR composite webpage should be a reliable path to them.

1.5 The JIR Client

The JIR server is designed as an API, which means that it is intended to be used by other programs that send information requests to it, and not for use directly be people. In order to facilitate the task of understanding how JIR works and to 'play with' it, we have developed a separate website called JIRCLI, for 'JIR Client' whereby the interested user can prepare requests to JIR and obtain a presentation of the results in a convenient manner. (The implementation of JIRCLI is actually part of Phase II of this project). The JIRCLI server is presently available at

http://piex.publ.kth.se:18089/

and it is also linked from the main JIR webpage mentioned above.

2 Acquisition and Maintenance of the JIR Knowledgebase

As Figure 1 suggests, the JIR knowledgebase for information about publishers' parallel publication conditions obtains its contents from two sources: from a downloaded copy of the Romeo/Sherpa database, and from specific documents that are posted or provided by individual publishers or journals. Both sources contribute to the same knowledgebase, and we shall first briefly describe its structure.

2.1 Knowledgebase Features

The following aspects of the publisher's parallel publishing conditions are represented in the JIR knowledgebase.

- whether the publisher will consider an article that has already been prepublished
- whether the publisher will accept postpublication or not
- whether the publisher requires separate payment in order to accept postpublication
- publisher requirements on embargo time, i.e., the required delay between publication of the article in the journal and its postpublication
- restrictions on the type of server (private, institutional, discipline-oriented, etc) that may be used for the above
- whether special rules with respect to the above aspects may apply for articles reporting work that has been funded by particular agencies. This includes both special rules made by the publisher, and special rules made by the funding agency
- which version of the article may be used, or may not be used in the case of postpublication, according to the publisher
- publisher requirements on what information must be included in the postpublished version of an article, or in web pages that serve as "wrappers" for such articles, or in web pages containing a list of such articles
- which graphical appearance of the article may be used, or may not be used, according to the publisher
- publisher requirements on what must be done to a prepublished version of an article at the time when the final article is published in the journal

All of these are actual restrictions that are being made by some publishers. This rule base has a considerable complexity, therefore.

Notice in particular the potential interaction between the requirements of funding agencies and those of particular publishers, and the need for advise to the authors about how to handle such conflicts. Notice also the complexity of rules that arises when different sets of rules apply depending on the funding agency, or depending on whether the author agrees to pay a parallel publication charge.

A more detailed description and the technical details of the data representation can be found on the JIR system website at http://www.ida.liu.se/ext/jir/.

2.2 Acquisition from the Romeo/Sherpa Database

The initial version of the JIR knowledgebase was obtained in the following way. The on-line Romeo-Sherpa database was downloaded and parsed, and unique identifiers were introduced for each one of the publishers listed. Each identifier was associated with the set of sentences indicating this publisher's parallel publishing conditions. Since identically the same sentence could occur repeatedly, we collected the set of different sentences and assigned an identifier to each. Sentences that were effectively synonymous were collected into equivalence sets. Each publisher identifier was therefore associated with a sequence of sentence equivalence sets.

After this, each sentence equivalence set was associated with a formal translation as a symbolic expression representing its contents in a knowledge representation that had been designed for this purpose. (The character of the information was such that it was more reasonable to use a specialized and ad-hoc representation, rather than using a textbook notation). The translations for a given publisher were combined so as to form a composite formal expression that would, ideally, express exactly the contents of the original set of phrases for the publisher.

The original sentences and the translations were inspected manually to check whether this process had worked correctly. It was clear that the initial process had to be modified in the following way: when several phrases and phrase translations addressed the same aspect of the parallel publishing issue, then in some cases they should be combined using an 'and', and in other cases using an 'or' connection. This choice depended on what issue was being addressed.

After this modification, the process was run through again, and a second manual check was made. The result was that in almost all cases, the meaning of the entire sequence of sentences for a publisher was correctly obtained by combining the translations of the constituent sentences, taking the distinction between 'and' and 'or' connections into account. Only a very small number of exceptions were found, and were treated manually.

The outcome of this process makes us confident that new and extended sets of Romeo-Sherpa-style indicative sentences can be processed in the same way, so as to keep the database up-to-date.

Phase I of this project has been working with an older version of the Romeo/Sherpa database. Catching up with the current database is one of the tasks for Phase II.

2.3 Acquisition from Original Publisher Information

Besides knowledge acquisition from Romeo/Sherpa, we also wish to establish a system for acquisition of original publisher information. As a first step in this direction, a testbatch of around 20 publishers were identified and studied in depth, first of all by inspecting their webpages in search for their standard IPR publication agreements. It was our original expectation that all publishers of some stature would provide this information. We were disappointed to find that this was not the case; the following are some of the reasons.

- Some publishers do not publish their standard contract on their webpage
- Furthermore, when we wrote to publishers to ask for a copy of their publication contract, some refused to release it
- In a number of cases it became clear that different journals in the same publisher use different publication contracts
- In some cases, the publisher's webpage provided indeed the text of their standard contract containing quite restrictive conditions, but complemented it with a separate page for, for example, "information for authors" where less restrictive rules were posted. In particular, acceptance of parallel publication was sometimes specified in these informal web pages although the formulations of the formal publication contract would not have allowed them.

This state of affairs is regrettable from the point of view of the research community and the open access movement. The open availability of publication contracts is obviously a necessity for the efficient and correct operation of parallel publication in an institutional repository, as well as for its support services such as the Journal Information Resource. Furthermore, the situation where some of the authors' rights are only specified informally is unsatisfactory since it leaves open the question of what will happen if the publisher should decide at some point to change its "information for authors," or if the publisher situation is changed by acquisition or merger.

Another unclear point that we observed in this context concerns the permission to post the published article on the website of "the author's employer." One would like to assume that this wording means that the permission applies to each author, even if there are several authors having different affiliations. However, it is not clear what happens if an author changes employer. Will this clause permit the article to remain on the site of the previous employer, or will it require the article to be moved to the site of the new employer? Conversely, will the author be allowed to post previously published articles on the website of the new employer? It would be useful to have a clarification on points such as these.

Based on preliminary observations such as these, we decided that it would be premature to set up an activity in Phase I towards acquiring additional parallel publishing information, besides what is already available through the Romeo-Sherpa database. We chose instead to give priority to an activity towards investigating, in a systematic fashion, what kind of information will typically be available from the publisher side, and how this information is typically represented and made available. The results of that study were reported in a separate memorandum, Reference 3.

3 Project History during Phase I

The initially set goal of the project was to develop a resource that could be accessed by institutional repositories and that would automate the answers to the type of queries that are today done manually using the Romeo-Sherpa website. A group was set up consisting of Erik Sandewall (investigator) and Gunnar Eriksson, Preben Hansen and Oscar Täckström at the Swedish Institute for Computer Science (SICS) for the purpose of performing this task. The work was organized so that Erik Sandewall and Gunnar Eriksson would work concretely on the system described in Section 1 of this report, and Preben Hansen would work on the topic of Section 2. Oscar Täckström is a Ph.D. student in computational linguistics, and was adjoined to the group because of our interest in a possible future extension in the direction of automatic or semi-automatic analysis of the contents of the publishers' copyright transfer contracts and their related documents.

According to the initial plans, it should have been possible to prepare a JIRtype system that relies on both the information obtained from Romeo-Sherpa and information extracted directly from publisher sources for a "testbatch" of publishers. However, as the work proceeded we came to the realization that both parts of the task were significantly more complex than what we had understood initially. This led to a restructuring of the project, a clearer separation of the two parts, and a proposal to extend the activity through a Phase II. We also reset the goals so that the implemented JIR system from Phase I would not rely on information from the testbatch used in the other half of Phase I; the convergence was postponed for Phase II.

The implementation of the JIR part also led us to realize the need for including the LABMOP concept in the overall design. We therefore decided to set up a testbench consisting of the JIR system being developed here, the previously existing MADMAN system for author support (MADMAN = Management of Articles, Data, Manuscripts And Notes), a newly implemented, ad-hoc LABMOP system, and a small mockup of a journal editorial system. Using this testbench which is called CAPPA, we were able to make a demonstration of the entire workflow both from the author point of view and from the repository point of view. This demonstration was made in February, 2009.

The demonstrated system used the two-pronged approach for access to Romeo-Sherpa data that was mentioned in Section 1. The direct, dynamic access to the Romeo-Sherpa site was implemented by Gunnar Eriksson; the use of the JIR knowledgebase was implemented by Erik Sandewall.

One result of the demonstration was that the representatives of the DiVA institutional repository system showed a clear interest in making use of this service, if it can be set up in a reliable form and with a better coverage of publishers and journals than what was the case for the demo. Given this, and given that some of the funds for Phase I still remained, it was decided to use the remaining funds and remaining time for making the existing system more stable, as a first step towards putting it in regular use. This work has continued during 2009 and has been done by Erik Sandewall together with Henrik Lundberg at Linköping University, and has been completed by the end of the year.

The work on acquisition of information about rights and conditions was performed by Preben Hansen and mostly during year 2008. One initial goal was to organize a repository of time-stamped IPR contract information, so that our document base in this respect would not only contain the relevant documents for each publisher under consideration, but also timestamped earlier versions of these documents as posted on the publisher's website. This could be of importance when a publisher changes their parallel publishing conditions.

A second goal was to make a "map" of what kinds of documents may be available and may be of interest as source information for the JIR knowledgebase.

In this branch of the project, it was soon discovered that the available information in the publisher's webpages is remarkably diverse and incomplete. Preben Hansen considered a testbatch of around 20 publishers, and made a careful analysis of the information that was available from them, resulting in a proposal for how to structure an archive of publisher's parallel-publishing-related documents. His work and the resulting proposal is reported in a working memo (Reference 3) which was reported at the same time as the system demo in February, 2009.

4 The System Demonstration and the CAPPA Testbench

The system demonstration and the testbench that was used for performing it have been described in an earlier report, Reference 5. The following are the highlights of that report.

4.1 Agent Structure

The CAPPA system is organized as a set of *software agents* that communicate with each other by message-passing. Some of the agents are also able to communicate directly with a user. This applies in particular for those agents that serve as "personal assistants" to their respective authors. The following agents were used for the demo. Each agent or agent type is described with the name that it has in the system.

author-agent	A software agent that is conceived as a software
	assistant for a particular person
labmop	A server agent for Management Of Publications

on the level of LABoratory or research group instarch A server agent for management of an INSTitutional ARCHive A software assistant agent for a journal editor editor-agent that makes decisions about the acceptance of contributed articles jes A software agent for a journal. We model it as a server agent and not as a software assistant, and assume that it will communicate with editor-agents, author-agents and publisher-agents. The initiative for these communications will be variable publisher-agent A software agent for a publisher of scientific journals. It is modelled as a server agent, like for journals. The officer in charge of a set of journals may require her or his assistant agents, but we leave this outside the model. jir A Journal Information Resource, that is, a server agent that provides information about publishers and journals, for example, their IPR conditions for pre- and postpublication.

Each agent maintains its own collection of information, for example as a database, and in terms of the model there is no common file storage or database for several agents. All information-sharing must therefore occur through message-passing which means that it is modelled explicitly. In our case the message-passing consists of sending a *command* and receiving back the *outcome* of its execution. We shall refer to this process as *command-passing*.

Command-passing activities between these agents always start with a user command to one of the software assistant agents. The execution of the command may involve the sending of one or more commands. When an agent receives a command it executes a *script* that specifies what it is required to do. This may involve sending additional commands to other agents, or informing its user in the case of software assistants. It may also involve sending an acknowledging message to the sender.

In general, a system architecture that is based on command-passing between agents will only work properly if there are one or a few *proactive agents* that check that each item in the flow moves forward (each article, in our case) and is not left behind. This role belongs to the author and the journal's executive editor, with the assistance of their respective software agents.

4.2 Action Sequence in the Author-side Workflow

The demonstration scenario defines one software agent for each one of the agent types shown above, except that three author agents are defined, each one serving one particular author. The following sequence of steps are used in the standard case where an article eventually gets published. Each step is effectuated by a command from the user to her software assistant agent, unless otherwise noted. It is assumed that this agent already contains the article and its initial metadata (authors, title, abstract, etc) when the sequence starts.

- *Register the article* with the laboratory-level agent, providing it with the basic information about authors and title.
- Request prepublication to be made by the laboratory-level agent (optional).
- *Register intention to submit* the article to a particular journal. This registration is again made with the laboratory-level agent. The author receives advise about submission procedure and gets access to help services for preparing the manuscript in the way required by the journal in question.
- Submit the article to the journal that was selected in the previous step. The software assistant of the corresponding author sends the manuscript and the required accompanying information to the journal agent, and receives a confirmation in return. It then informs the laboratory-level agent that this has happened.
- *Editor in charge* of the article specifies the acceptance decision to his or her assistant agent, which sends this information to the journal agent. This agent in turn informs the assistant agent of the corresponding author, which gives the information to its user and informs the laboratory-level publication manager.
- Corresponding author proposes postpublication of the article to the institutional archive agent. This agent inquires the laboratory-level agent about publication details, and then makes its decision about whether, when and in which appearance the article is to be posted on its public server.

Each step in this sequence requires that certain basic data about the article shall be transmitted, and several of the steps also require the transmission of full-text files. The command-passing shown above is merely a skeleton for the flow of events, and many of the steps require a number of things to be done, such as reformatting the article with additional information required by the publisher, selecting a web address (URL) or other data item that can serve as a persistent identifier of the parallel-published article and generating appropriate initial contents for it, and so forth.

Likewise, what is shown as one step in this sequence may actually consist of several interactions, for example, for the choice of journal. Interactions between co-authors of an article are not modelled in this sequence but they should be an important part of any author-support system that is used when co-authors work at different locations.

4.3 Operational Use

Once the types of IPR information has been identified, it follows fairly directly how it has to be used in the publication-domain architecture. Its operational use can be considered from the point of view of the Institutional Repository, and from the point of view of the author and the proposed LABMOP system.

The specific institutional repository usage has already been discussed in the connection with Figure 1. One may add, however, that there remains an important question concerning whether a system such as this shall be allowed to operate entirely automatically, or whether a final manual check of the conditions will always be necessary. This hinges on the question what will happen if and when an institutional repository accidentally does parallel publication of an article in violation of the author's publication contract. In an "understanding" mode, the publisher will inform the Repository of the situation and the Repository will retract the article from its server. The Repository operator has acted in good faith and in accordance with best practice, and no further action is taken.

In an "aggressive" mode, on the other hand, the publisher will take legal action merely because the Repository has posted the article. In the extreme, a "three-strike" policy along the lines of the French HADOPI legislation could threaten to

close down a university's Internet connection, thereby making parallel publication effectively impossible. This is not a likely development, of course, but it illustrates the importance of paying attention to the possibility and the consequences of mistakes in the handling of the rights and the conditions for parallel publication.

With respect to the user perspective on the use of a Journal Information Resource, we propose that there are three important kinds of uses:

- *Early warning* in order to inform or remind the author of restrictions that may be relevant for his or her decisions about an article. For example, if the author registers the intention to submit a previously prepublished article to a journal that does not accept such submissions, the author shall be given a warning. The same applies if the author has submitted the article to such a journal and then proposes prepublication during the reviewing period
- *Impose restrictions on actions,* in particular, prevent the laboratory-level or institution-level archive from posting an article in cases where this is inconsistent with known agreements between author and publisher
- Information formatting actions, including in particular the reformatting of an article in order to comply with publisher requirements (as well as institutional requirements, in fact), and the generation of web pages that present an article or a set of articles

Additional work will be required in order to implement the forms of interaction between the system and the authors of articles in order to communicate early warnings. The other two types of uses are internal to the Institutional Repository, or the LABMOP if applicable, and can be implemented within these systems without undue difficulty.

5 Reorganization of the JIR Software

The system demonstration and the use of the CAPPA testbench was effective for helping us to understand the design issues in a sufficiently broad perspective, and it also identified some weaknesses in the software implementation of the JIR system being used there. It was felt necessary to correct these before proceeding. The last part of the Phase I of this project, during year 2009, was therefore used for revising the software and rewriting some parts of it. This has led up to the version of JIR that is now available on its website as described in Section 1.4.

6 Conclusions

The work in Phase I has achieved the following:

- A first version of the Journal Information Resource (JIR) has been implemented, including both its knowledgebase and the software that applies it to specific queries. The software design has been iterated in order to obtain a robust implementation, and is available on a public website for experimentation. The knowledgebase has a well-tested structure but its contents are not up-to-date, leaving further work for Phase II.
- The design for this JIR has been given preliminary validation by embedding it in the CAPPA testbench, together with testbench versions of related software systems. The resulting software systems aggregate has been tested by ourselves and has been shown in a demo session for the sponsors and other interested parties. This gives us confidence in the appropriateness of the chosen design, and it has also given suggestions for certain easily feasible improvements.

• Concurrently with this, we have investigated the structure of contractual information that may be available from the side of the publisher, and planned for a systematic acquisition, archiving and analysis of that information.

This constitutes a natural baseline for the continued work during Phase II of the present project.

Acknowledgements

The support of the OpenAccess.se development program for this work is gratefully acknowledged.

The availability and free use of the Romeo/Sherpa database has been essential for the present project, and our indeptedness is acknowledged with deep appreciation.

The work described here has been performed by Gunnar Eriksson and Preben Hansen at the Swedish Institute of Computer Science (SICS) and by Henrik Lundberg at Linköping University, besides by Erik Sandewall (KTH and Linköping University) as project leader.

References: Project Reports

1. Erik Sandewall: OA-publicerade domänmodeller avseende vetenskaplig publicering och gruppstruktur. (OA-published domain models concerning scientific publishing and research-group structure). Organizational Memo number 4, Division of Publication Infrastructure, KTH, Stockholm. Available at http://piex.publ.kth.se/reports/rapp/004/.

2. Erik Sandewall: Delrapport för OpenAccess.se-projektet 'Domänmodellering av rättigheter och bivillkor vid parallellpublicering av vetenskapliga artiklar'. (Intermediate report for the OpenAccess.se project 'Domain modelling of right and constraints for parallel publishing of research articles'). Organizational Memo number 7, Division of Publication Infrastructure, KTH, Stockholm. Available at http://piex.publ.kth.se/reports/rapp/007/.

3. Preben Hansen, Gunnar Eriksson och Oscar Täckström: Steps towards automatic acquisition and recognition of IPR conditions for parallel publishing. Forthcoming project report from the Swedish Institute of Computer Science (SICS); preliminary version is available.

4. Erik Sandewall: Support for Managing IPR and Parallel Publishing in the MADMAN Research Author Support System. Report number 2, ADEPT project, Division of Publication Infrastructure, KTH, Stockholm. Available at http://www.ida.liu.se/ext/caisor/pm-archive/adept/002/.

5. Erik Sandewall: Demonstrating the Use of Author-Deposit Restrictions in Publication-Related Software Systems. Report number 4, ADEPT project, Division of Publication Infrastructure, KTH, Stockholm. Available at http://www.ida.liu.se/ext/caisor/pm-archive/adept/004/.