Intellectual Property Rights Management with Web Service Grids

Christopher Cieri email: ccieri
Denise DiPersio email: dipersio

Linguistic Data Consortium
3600 Market Street
Philadelphia, PA. 10104
email: @ldc.upenn.edu

Abstract

This paper enumerates the ways in which configurations of web services may complicate issues of licensing language resources, whether data or tools. It details specific licensing challenges within the context of the US Language Application (LAPPS) Grid, sketches a solution under development and highlights ways in which that approach may be extended for other web service configurations.

1 Introduction

Growing interest in web service architectures raises questions about how such uses of language technologies and other resources interact with licensing constraints, including those that were imagined at an earlier time when resources and tools were more likely controlled by individual user organizations. Research communities that depend upon language resources (LRs) have become accustomed to, if not delighted with, the need to agree to certain limitations on the use of such resources. However, historically, negotiations concerning the use of LRs occur relatively infrequently. Even the largest data centers produce only a few new resources each month, generally under one of a small number of familiar license types. Once the resource has been acquired, integrating it in a local workflow requires time, creating a natural brake on the need to acquire new resources. Grid infrastructure, on the other hand, promises the ability to very rapidly build pipelines from existing services and resources. The common vision of web service architectures is that they reduce the burden of tool integration by presenting the tools as services and coordinating their input and output requirements. However, absent a similar mechanism for coordinating the licenses that constrain LR use, Grid operators risk creating infrastructure that simultaneously ameliorates the tool integration problem while exacerbating the licensing problem. In the sections that follow, we describe an approach that addresses the general problem of documenting, communicating and partially enforcing licensing constraints within a service grid.

2 Web Section Complexities

Human Language Technology related web services, singly and in various configurations and clusters, constitute a new ecosystem in which LRs, specifically data sets and tools, may be implemented and combined in ways not necessarily contemplated by existing intellectual property law and contracts that apply to the web services' constituent components. For example, traditional licenses may permit distribution from a data center to a licensed user organization and processing by either, but may prohibit distribution from the user to any additional parties. Even if it were clear that this constraint were intended only to block redistribution to unlicensed users, it is not clear whether all copyright holders would agree that moving the same data over the web to be processed by web services should be allowed. Another example involves the attribution and license requirements of shared software. In the past, licenses were typically described in a document included with the software source code. Attribution requirements were satisfied by listing software authors' names in similar documents or by displaying them in a header presented when the software was invoked at the commend line. However, users of web services may never see a source code repository or a command line.

Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT, pages 93–100,
Dublin, Ireland, August 23rd 2014.

Service grids, as just one of multiple possible configurations of web services, have different stakeholder types: grid operators who maintain the software and servers that allow service registration and discovery; two types of service providers, those who provide access to data and those who provide access to software; and of course users. In addition to multiple stakeholder types, such Grids also have multiple instances of users, data providers and software providers. Where grids have been federated, there are also multiple grid operators. Each of these stakeholders may have different desires relative to the behavior of web services, most importantly where intellectual property protection is concerned. Beyond the obvious conflict that users typically want fewer restrictions on resources than providers, grid operators or service providers may require compensation, grid operators may wish to track and record user behavior, service providers may demand attribution, limit use of their services to the non-commercial sectors and may wish to exploit data that passes through their service nodes for purposes of further system development or evaluation.

In addition to multiple stakeholder types and multiple instances of each, grids also combine these data and software services in various combinations that affect licensing in a variety of ways. Figure 1 summarizes three simple cases. In the first, users direct data they own or control through an external service controlled by a second party. In the second case, both the data and the processing are controlled by a single entity who is not the user. In the third case, one external party controls the data while another controls the software. In each of these cases, the interaction of multiple parties may complicate licensing by introducing new and idiosyncratic constraints.

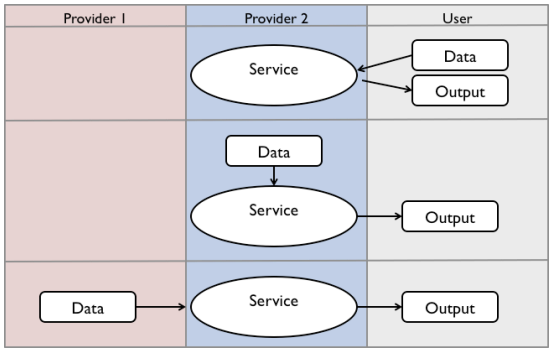

Figure 1: Simple Configurations of Web Services

Figure 2 sketches more complex cases in which data, controlled by the user or not, passes through multiple services controlled by independent parties. Examples of the first two use cases might include speech translation, configured as speech transcription followed by translation of the transcript text, in which the input speech is controlled by the users (e.g. in voicemail transcription) or is controlled by an independent party (e.g. translation of broadcast news). In the third case, not only are there multiple services operating on the same data, but these services are configured as generic engines that require models provided by other parties to operate on specific languages. One example might be language identification engines that accept new models in order to recognize new languages.

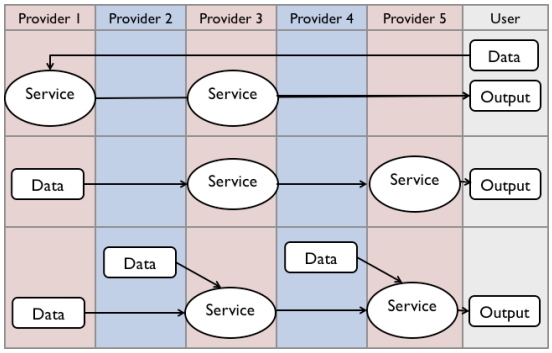

Figure 2: More Complex Web Service Configurations

Web service grids further complicate licensing because no provider or user controls the entire ecosystem. Grid operators, software service providers, data providers and users may all be distinct from

one another. Indeed in the future imagined by grid proponents, there are many software and data providers and even more users on any one grid, and many grids federated to permit users of one to access services on another. In such an environment, it is clear that each user, provider and grid operator may act independently and sometimes at cross-purposes to others.

3 Approaches to Grid Licensing

One can imagine multiple approaches to harmonizing this ecosystem. First, one might choose to try to constrain service and data providers insisting that as a condition of participation in the grid, they must agree to make their services available to specific classes of users under pre-defined terms. The NICT Language Grid did this by establishing a Service Grid Agreement. However, it is equally possible to construct a grid in which service providers are not the owners or developers of all the software they use to provide services. For example, within the US NSF-funded Language Application Grid, two of the principle service providers, Brandeis University and Vassar College, have created services based on third party software such as NLTK (Bird, Klein, Loper 2009) and the Stanford Toolkit (Manning et al., 2014). In this case, the service providers do not own the tools and thus cannot enter into agreements about the terms under which the software underlying their service may be used. As a solution, one might imagine providing software services under whatever licenses the underlying software has imposed and then constrain the users to comply with these terms. A third alternative would be to assume that all parties take responsibility for their own actions during grid operations and to impose no controls over either providers or users. The fourth option is of course to constrain both providers and users. In this paper we will argue for this last hybrid.

Descriptions of the licensing approaches used by existing grids are scant, only rarely presented in published works, occasionally described on project web pages and sometimes left to be understood from the licenses used. Piperidis (2012) describes META-SHARE as a membership based infrastructure in which resources are available under one of four license types. While META-SHARE encourages within-network sharing with the fewest constraints possible, the four license types permit a range of constraints including those expressed by the Creative Commons[1] licenses and also allow for fees. META-SHARE servers harvest licensing elements from contributed services and present them to users as a table, in fact the model for our Table 2 below. The Language Grid[2] developed by NICT, includes license text, where available, in the description of each resource it provides. The Language Grid also provides a tool for composing workflows. Upon execution of a given workflow, the Grid displays the sequence of licenses that affect the use of the workflow component tools and data. Bosca et al. (2012) describe LinguaGrid[3] as "open to different operators (Universities, Research institutes, Companies) with configurable services access policies: free, restricted to registered users, research or commercial licensing". LinguaGrid is built upon the grid infrastructure developed by NICT and presumably uses the same approach to license management. CLARIN[4] documentation describes a rich set of licensing options for service providers. Many have equivalents in the Creative Common licenses though CLARIN enriches this set by allowing providers to require that published papers based on CLARIN resources are reported to the providers and that derived resources are deposited to CLARIN, a specific variant of the share-alike constraint. CLARIN also provides a legal help desk to answer questions about licensing among other issues. The LAPPS Grid has developed an explicit model for license management that is membership based, allows for a wide range of license types and fees, presents a summary of license constraints and actual licenses prior to workflow execution and even prevents the subset of license violations that can be detected at execution time.

Grid architecture constitutes a new approach to combining LRs. Providing clear documentation of the terms under which providers and users operate offers peace-of-mind to resource providers and clarity to service users either of which group may otherwise opt out of an initiative whose risks are incompletely understood. Furthermore, it is probable that for service providers who do not own the underlying data or software, imposing constraints on users may be not only a wise idea, but also a legal or contractual obligation. It is important to note that many license terms constrain behavior that

[1] http://creativecommons.org/
[2] http://langrid.org/en/index.html
[3] http://www.linguagrid.org/
[4] http://www.clarin.eu/

may occur long after web services have run, for example commercial use of output. Therefore, grid operators are not in a position to strictly enforce license terms. They may however, block obvious and immediate violations of licenses, make users aware of constraints that affect behavior and secure their agreement to relevant terms.

4 Dimensions of Constraints on Language Resource Use

The licenses that constrain the behavior of language resource users, and thus grid users, vary along a number of dimensions, the first of which pertains to the object being licensed. Software licenses typically constrain the use of software and derivative works. Data licenses similarly constrain the use of the data and derivative works. However, derivative work, to the extent that the term is defined at all, seems to refer to other data in the case of data licenses, and to other software in the case of software licenses. Importantly, none of the software licenses reviewed for this paper made a clear attempt to constrain the use of their output, which is often data, while many data licenses do constrain the use of processed data.

The LRs used in web services may be owned by the user, may be in the public domain or may be copyrighted by someone other than the user. Copyrighted LRs may be constrained as to use or as to user. The commonest use constraints typically prevent commercial use and the creation or commercial use of derivative works. They may prevent distribution of the LR or derivative works or they may require that products whose creation relied upon the licensed resource be shared under the same terms (also known as the Share Alike or Viral Copyleft constraint). They may require that users provide attribution of LR creators and/or cite the resource or a specific reference paper. Finally, any license may include other terms that have not been described here because they constitute the long tail of uncommon constraints. To give just one example, we are aware of at least one corpus that requires that recipients receive certification from their local Institutional Review Board that they have been trained in the treatment of human subjects.

An additional complexity in licensing constraints defined by use is that neither copyright law nor the software or data licenses reviewed for this paper provide a bright line to distinguish derivative works (which are typically constrained by such licenses) from transformative uses (which are typically not). Within HLT, we can imagine simple and stereotyped cases. Given one hour of audio recorded from a copyrighted news broadcast, the transcript of the audio and its translation into any language are derivative works subject, at least in the US, to copyright and any licensing constraints imposed on the audio. On the other hand, a unigram frequency list based on the transcript or translation is a highly transformed work generally considered immune to those same limitations.

License constraints related to the user, rather than the use, typically prevent commercial organizations from accessing the resource. In at least some cases, the intent of this constraint is to encourage potential commercial users to negotiate directly with the LR provider for access under terms that include a fee structure. More generally, the user types distinguished by LR licenses include academic and not-for-profit organizations, governments and commercial organizations. In some instances, companies engaged in pre-commercial technology development may be treated differently. In addition, a model of licensing constraints must distinguish organizations that have executed a specific license required by a LR from those that have not. Organizations may be licensed by enumeration or by features. As an example of licensing by enumeration, the Linguistic Data Consortium maintains databases of all users, all licenses required by their LRs and a table of which user organizations have executed each license. However, users may also be considered licensed if they possess certain features, for example, if they are non-profit organizations.

One use that seems to have been overlooked by existing grid licenses is the ability of service providers to derive benefit from the processing they offer. For example, one could imagine a translation service that not only outputs translations for submitted input text but also computes n-grams from that text and uses them to improve its source language models. Were this practice allowed, it would further complicate licensing within web service architectures where it is not always clear that the user who submits data for processing has the authority to permit the service providers to exploit that data.

5 Combining Licensing Constraints

Having enumerated the dimensions along which LR use may be constrained, we can easily imagine some use cases in which specific workflows should be prohibited or at least flagged. The obvious case would be one in which some input data required a specific license that the potential user had not executed. Another example would be the case in which some processing service required a fee that the potential user had not yet paid. Similarly a commercial organization seeking to process data that is only available under a no-commercial-use license should be prevented or at least warned by the service grid. At least within the United States – and this probably holds for many other jurisdictions – the law that governs copyright and the individual licenses commonly associated with LRs are relatively underspecified on a number of questions relevant to web services. For example, within US copyright law the only functional definition of "fair use" is a description of the four dimensions along which fair use claims are to be evaluated. Given this situation, we should not be surprised that current bodies of law offer no calculus for combining constraints imposed on the multiple LRs that may support any given workflow.

For example if a specific pipeline makes use of two data resources, one of which permits commercial use while the other prohibits it, what constraints apply to the final output? To make this concrete, consider a pipeline in which a language identification service detects the language of an input text and routes it to an appropriate machine translation service. If the language identification service relies on a data set available under a no-commercial-use license but neither the input text nor the translation engine are similarly constrained, may the user sell access to the translation? Our tendency may be to think this use is acceptable. Would we feel the same if the translation engine relied on data that imposed the no-commercial-use constraint? What if the input text was available under a no-commercial-use license but no other component in the pipeline constrained use? While we may have intuitions about acceptable use in these cases, there is no body of law, nor much precedent, to support one or another interpretation.

6 The Language Application Grid

In order to work through a possible solution, we now consider the specific tool and data resources implemented in the US NSF funded Language Applications Grid. To date, the LAPPS Grid has used 27 unique software packages (programs, toolkits, APIs, libraries) that are available under the 9 unique licenses summarized in Table 1.

Table 1: LAPPS Grid Software by License

License	Software
Apache 2.0	Language Grid software, NLTK, ANC2G0, UIMA, OAQA, Uimafit, guava-libraries, ActiveMQ, AnyObject, Jaxws-maven-plug-in, Jetty, OpenNLP
BSD	Hamcrest, NERsuite, CRFsuite (in NERsuite)
CDDL 1.1	Jaxws-rt
CPL 1.0	MALLET, AGTK, JUnit
Eclipse 1.0	logback (v1.0), Jetty
HTK-Cambridge	HTK
MIT	Mockito, libLBFGS (in NERsuite), GIZA (v3)
Python	NLTK
WordNet	Genia tagger library (in NERsuite)

Many of the constraints imposed by these licenses fall into recognizable categories summarized in Table 2

Table 2: LAPPS Grid Licenses and Common Constraints

License	Redistribution	Use	Derivative Use	Attribution	Share Alike	Fee
Apache 2.0	Yes	Commercial	Commercial	Yes	No	No
BSD	Yes	Commercial	Commercial	No	No	No
CDDL 1.1	Yes	Commercial	Commercial	Yes	Yes	No
CPL 1.0	Yes	Commercial	Commercial	No	No	No
Eclipse 1.0	Yes	Commercial	Commercial	Yes	Yes	No
HTK-Cambridge	No	Commercial	Commercial	No	No	No
MIT	Yes	Commercial	Commercial	No	No	Yes
Python	Yes	Commercial	Commercial	Yes	No	No
WordNet	Yes	Commercial	Commercial	Yes	No	No
LDC FP Member	No	Commercial	Commercial	No	No	No
LDC NFP Member	No	Research	Research	No	No	No
LDC Non-member	No	Research	Research	No	No	Yes
CC-Zero	Yes	Commercial	Commercial	No	No	No
CC-BY	Yes	Commercial	Commercial	Yes	No	No
CC-BY-SA	Yes	Commercial	Commercial	Yes	Yes	No
CC-BY-ND	Yes	Commercial	None	Yes	No	No
CC-BY-NC	Yes	Research	Research	Yes	No	No
CC-BY-NC-SA	Yes	Research	None	Yes	Yes	No
CC-BY-NC-ND	Yes	Research	None	Yes	No	No
GPL (v2,3)	Yes	Commercial	Commercial	Yes	Yes	No

These many license have in common a relatively small number of constraint types and values as summarized in Table 3.

Table 3: LAPPS Grid Common Constraints and Values

Constraint	Values
Redistribution	Yes/No
Use	Commercial/Research Only
Derivative Use	Commercial/Research Only/None
Transformative Use	Commercial/Research Only /None
Attribution	Yes/No
Share Alike	Yes/No
Fee	Yes/No
Other Specific License, Constraint	--

However, as one considers the complexity of licensing with the grid, it is important to also consider the limitations on the role of grid operators relative to prior practice. Traditional language resource distribution, before the era of web services, treated licensing constraints variably. For example, where users are required to pay a fee in order to access a LR, that fee is normally required in advance. If a resource requires a specific license to be executed, some data providers may withhold the resource until the agreement is signed either on paper or via a click-through agreement. Others may provide the license with the LR and a statement that by accessing the data, the user is agreeing to the terms of the license. However, beyond these cases, there is little attempt to block access to a resource until licensing terms have been satisfied. Indeed many licensing terms constrain future action and thus cannot be required as a condition of access. For example, the constraints on redistribution, use of derivative works, attribution and share alike all affect action that necessarily takes place after the LR has been accessed. Given these limitations, we expect that the ability of any web service policy or procedure to enforce such constraints is similarly limited. Thus, within the LAPPS Grid, we distinguish two types of enforcement of licensing constraints, requirement and notification as summarized in Table 4. In a

small number of cases, we block the execution of a service pipeline if required conditions are not met but otherwise accumulate notifications that we present to users before allowing them to execute the pipeline. We must also note here that no summary of licensing terms can legally stand-in for the actual license executed so that any approach we use must also make reference to the actual licenses.

Table 4: Constraint Enforcement

Constraint	Action
Redistribution	Notify
Use	Notify
Derivatives Use	Notify
Attribution	Notify
Share Alike	Notify
Fee	Require
Other Specific License	Require
Other Specific Constraint	?

7 A Grid Licensing Model

Putting together the discussion to date, we propose the model in Figure 3 for managing licenses within a service grid framework. Specifically, this model benefits from features already implemented for the LAPPS Grid while imposing some limitations of its own. First, within the LAPPS Grid, users build pipelines using one of two workflow management tools developed by the project. The *Composer*, described in greater detail in Ide et al. 2014, displays for the user the set of available tool and data services allowing the user to select one or more, determine their order of application and even create branches to allow two or more tools of the same types to operate on the data in parallel so that their performance may be evaluated and compared. The Composer takes note of the input and output requirements of each tool in the chain and, in complex workflows, correctly routes data to appropriate processing services. The workflow *Planner*, still under development, allows the user to specify input data and desired output and then uses its knowledge of each tool's inputs and outputs to construct a pipeline that produces the desired result. The licensing model makes use of these workflow managers. First we require that any service registered in the grid only respond to requests from one of the workflow managers. This keeps the grid ecosystem closed and prevents a user from directing output of one of the services outside the grid where one cannot monitor use. We also require that service providers register the licenses that govern use of their services. The user initiates a session with the workflow managers by authenticating themselves. As the user builds a workflow, the manager requests from each service the list of constraints imposed. As in Table 2 and Table 4, these may be requirements for a fee or the execution of a specific license or they may be notifications of the future behavior expected of users. The workflow manager also queries a local database or API connected to a service provider or data center to determine whether the user has satisfied the payment and specific license requirements. If not, the pipeline is blocked. Otherwise the user continues to build the pipeline while the manager accumulates a summary of the click-through licenses required and general licensing constraints imposed. Before the user can execute the pipeline, the workflow manager presents a summary of the licenses required, with links to the original text, as well as a summary of the general constraints imposed. The user must click to agree to the terms before processing will begin. For each service that provides processing, the workflow manager also displays any attribution requirements or license statements normally displayed at the command line or in a README file since these are generally invisible to a grid user.

Of course, this model only works if the grid or other collection of web services constitutes a closed system where a small number of management programs can control the inputs and outputs to each process. Naturally, the grid licensing model is unable to resolve issues that remain unresolved in general such as the lack of a bright line distinguishing derivative and transformative use of linguistic data and tools. In such cases it takes a legally conservative approach, acting for example as though as uses may be considered derivative and issuing appropriate warnings.

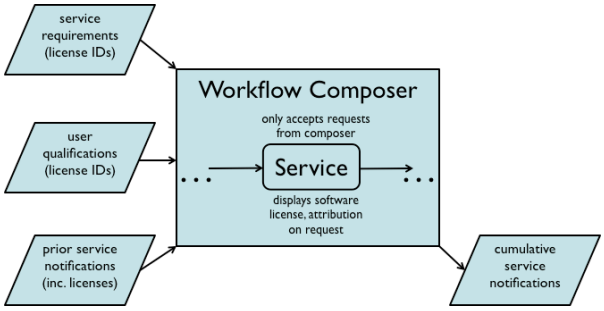

Figure 3: Service Grid Licensing Model

8 Conclusion

We discussed the features of web service architectures that complicate the licensing of LRs, including data and tools, both by introducing ecosystems not contemplated during the drafting of relevant intellectual property law and the development of the LR and by creating complex workflows not entirely under the control of any single user. We sketched the dimension along which licenses constrain LR use. Making the discussion more concrete, we then enumerated the license and constraint types that affect the resources used to build the US LAPPS Grid. Finally, we sketched a model for protecting intellectual property via the use of workflow managers while allowing users with appropriate credentials to construct complex pipelines. This approach relies on the closed nature of the service grid and would need to be extended in cases where the pipeline could combine web services without bounds.

9 Acknowledgments

This work was supported by National Science Foundation grants NSF-ACI 1147944 and NSF-ACI 1147912.

References

Bird, S., Klein, E., Loper, E. (2009) Natural Language Processing with Python. O'Reilly Media.

Bosca, A., Dini, L., Kouylekov, M., Trevisan, M. (2012) Linguagrid: A network of Linguistic and Semantic Services for the Italian Language. In Proceedings of the Eighth International Language Resources and Evaluation (LREC12), Istanbul, Turkey. European Language Resources Association (ELRA).

Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., DiPersio, D., Shi, C., Suderman, K., Verhagen, M., Wang, D., Wright, J. (2014) The Language Application Grid. In Proceedings of the Ninth International Language Resources and Evaluation (LREC14), Reykjavik, Iceland. European Language Resources Association (ELRA).

Ide, N. and Suderman, K. (2014). The Linguistic Annotation Framework: A Standard for Annotation Interchange and Merging. Language Resources and Evaluation.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D. (2014) The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.

Piperdis, S. (2012). The META-SHARE Language Resources Sharing Infrastructure: Principles, Challenges, Solutions. In Proceedings of the Eighth International Language Resources and Evaluation (LREC12), Istanbul, Turkey. European Language Resources Association (ELRA).