Mapping Localisation Workflows on the Cloud

Lamine Aouad
Localisation Research Centre
CNGL - University of Limerick
Ireland

Abstract

Today, traditional batch translation processes are no longer able to deal with the mass amount of available content. Automated management has become unavoidable. Recent advances in language technologies promise a variety of benefits and support to this automation. On the other hand, from the infrastructural perspective, Cloud computing and service-oriented modeling and architecture provide flexibility in integrating, developing, and running a variety of localisation and translation web services, which will make them readily available to translators and localisors throughout the world. However, the quality of the localisation and translation processes is an important matter and a challenging issue in this case. This depends on multiple parameters, including cost, deadlines, availability, quality of relevant linguistic assets, etc. In that aspect, we present a mechanism that is able to take into account predefined requirements and quality parameters to efficiently map an abstract workflow to the set of available service instances in the Cloud. We will also describe general issues facing automated localisation composition.

1 Introduction

Translation and localisation projects are already being built by large collaborations of components and services typically provided by translation agencies, language providers, freelancers, or even involving internal or external crowd collaboration in an increasing number of projects, commonly referred to as crowdsourcing (many companies have already adopted it in various tasks, Facebook for its UI, Microsoft for products terminology, among others. On the other hand, globalisation does not only relate to companies looking to get their products and services to an international audience, but also on how to address the issue of global information imbalance, breaking language barriers, and bridging cultural divides.

Many challenges are facing this kind of *large-scale* translation and localisation projects including the management of the huge volume of available content, which is continually evolving, and also personalisation, i.e. how to model users in order to address their specific needs in their languages. Traditional professional batch activities in localisation cannot deal with this large and dynamically composed content. Automated management and localisation have become unavoidable and a foremost necessity. Recent advances of technologies in natural language processing (text analytics, MT, etc.) have also made it possible.

From the computational and infrastructural point of view, the Cloud has some attractive qualities in dealing with such a scenario. It delivers storage and processing as a service, which means more flexibility. Translators and localisors will have access to their projects from just about anywhere by just logging in. Similarly, projects can access any of the available localisation web services spread all over the world. The Cloud is already making large scale and high-performance computing more easily available to researcher in many areas, with a reduced cost and computer expertise. The Localisation Factory as envisioned by the CNGL (Centre of Next Generation Localisation) (1) will be supported by the Cloud in making data and NLP web services readily available, allowing users to develop flexible and reusable localisation and translation services compositions.

In this context, the user needs to be able to state its QoS (Quality of Service) requirements which have to be taken into account for the definition of the concrete set of services, i.e the executable workflow composed of instances, as opposed to the abstract workflow, which only describes the components type and reflects the data dependencies. Some quality parameters are generally included in the business contracts as terms upon the workflow initiator and service providers agree. These contracts, or SLAs for Service Level Agreements, refer to an agreement regarding the guarantees of a web service, which include a set of service level indicators, among other components regarding the purpose or restrictions for the service instance. Each instance candidate is annotated with quality of service information that will be used during the selection process. This set can be refined and changed in order to express other features of the services such as the reputation, linguistic assets quality, or users rating.

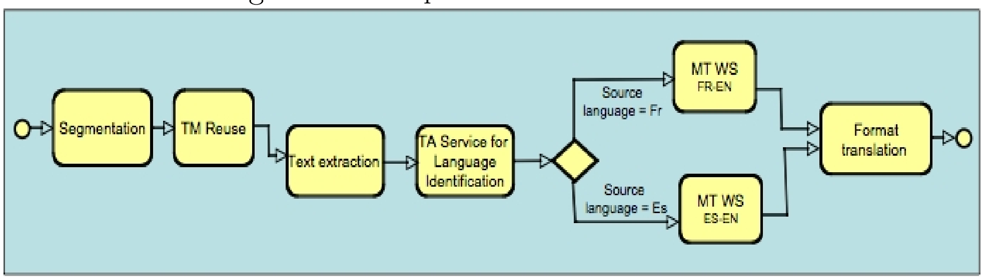
Here, we are naturally interested in the case where web services are exposed access points of localisation processes carried out by service providers or any third party organisation. A localisation process is composed of one or many web services. For instance, a process that enable translation may invoke web services from MT, TMs, parallel corpora, or QA checker web services. Picking up a suitable set of services to create localisation functions, addressing users needs, is not a trivial task as it is mainly based on dynamic attributes values. This study aims at proposing a mechanism in selecting the most suitable set of services and help the user to better review the services. This could also be applied to human interactions involved in the workflow composition. We will also outline some of the other challenges facing automated localisation composition.

2 Composite Web Services

Web services are Internet based applications that offer data or functional services programmatically, to other applications or the end user. They have recently emerged as de facto paradigm in implementing business collaboration processes within and across organisations boundaries. The general aim is to integrate applications by publishing, locating and invoking services all over the Web. These collaborations are expressed by a range of business process modeling languages tailored for web services (BPEL, Microsoft's XLANG, IBM's WSFL, BPML, etc.), and a set of supporting middleware using a stack of standards (including HTTP, XML, SOAP, WSDL, and UDDI). Details about these languages and standards is out of the scope of this paper, we refer the interested reader to (2) or (3).

The objective is to make full and efficient use of WS standards and Service-Oriented Architectures in wrapping today's NLP software and technologies. These services can then be composed and integrated into more meaningful localisation applications using the explicit control and data orchestration within process modeling languages, which make use of concepts from workflow management systems for web services composition. This composition can then make use of any potential exposed service, providing a high degree of flexibility in the integration process. This is already in use within the CNGL using the Business Process Execution Language (BPEL). It has also been used in other projects including the LanguageGrid project (4).

Figure 1: Example of WS use in Translation.



An example of a typical use of machine translation is given in Figure 1. This involve segmenting the content, looking up a TM for previously translated segments, and passing the rest to MT services. This would be a part of a more larger localisation workflow integrating interaction with professional translators for proof-reading and post-editing for instance. When many web services with similar capabilities are available on the Cloud, the choice of the most suitable provider is an important matter. The quality of service will distinguish these providers from each other in the creation of new localisation workflows.

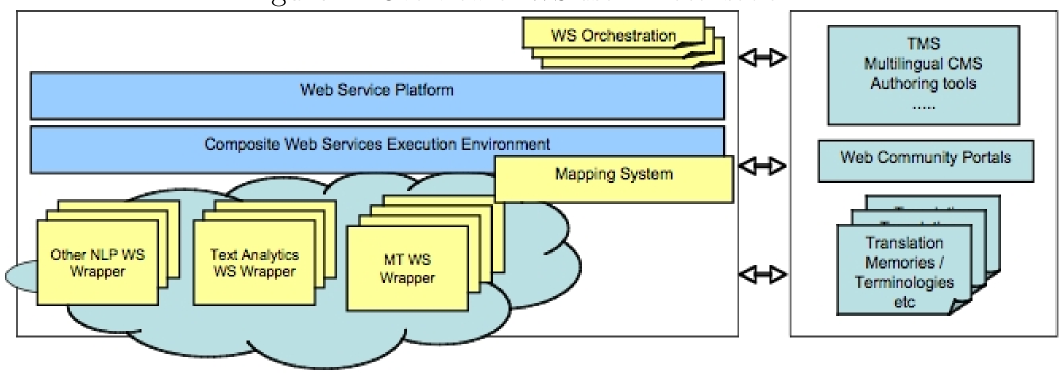
Currently, the OASIS TWS (Translation Web Services) TC is the only attempt to lead the localisation and translation industry to define a standard business process terminology which will drive the development of description files and publishing standards of localisation and translation WS. It also aims at defining service types that are relevant to the localisation and translation industry. It does not, however, intend at providing support for services negotiation or the monitoring / control of QoS indicators. We feel that this is of prime importance and need to be addressed. This study provides blueprint for building an efficient monitoring and mapping system in localisation WS.

3 Service-Oriented Localisation Factory

Traditional translation processes are no longer able to deal with the huge growth in volume of data. A leading translation market research firm[1] claims that only 0.5% of what needs to be translated is actually being translated

[1] Common Sense Advisory.

Figure 2: Overview of WS use in Localisation.

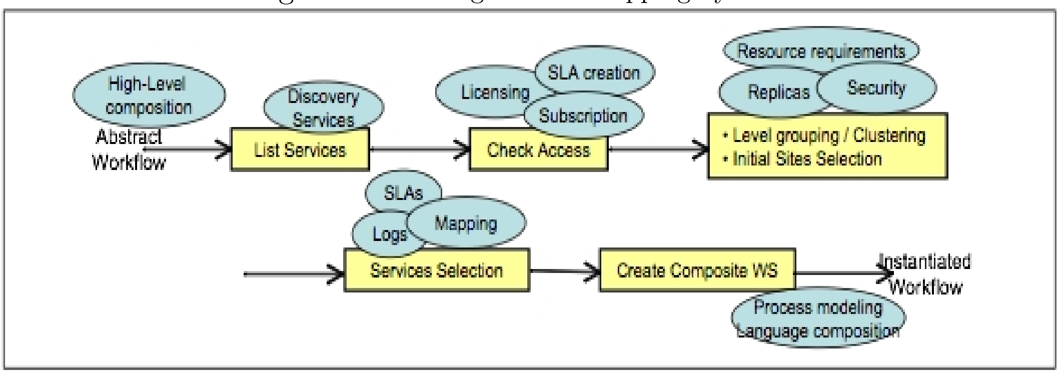


today. Essential information in health care, education, community support, etc. are available but not accessible to a vast majority of people in their own language. Advances in NLP technologies and especially MT research have made high quality translations possible in many language pairs. Billions of words are being translated every year by multinational corporates and NGOs using MT (Oracle, IBM, Asia Online, among many others). Also, translation and localisation projects usually involve multiple and complex set of tasks carried out in different locations by different services or companies. The management is a very complex task. Automation and integration of all these processes (i.e. NLP services and the data flow) is now possible using a Service-Oriented Architecture within what we term the *Next Generation Localisation Factory (NGLF)*.

On the other hand, in addition to the already mentioned TWS, many standards exists in the localisation industry to support the interoperability between the components. LISA (Localisation Industry Standards Association) and OASIS (Organisation of the Advancement of Structured Information Standards) have developed various standards, including TMX, TBX, XLIFF, among others. Some of these standards are more supported than others, but in general a lot more effort is needed in terms of integration, understandability, and interoperability of localisation and translation data, processes, and technologies.

A general overview on the use of web services within current practices in performing localisation and translation workflows is given in Figure 2. This would include a TMS system for TMs integration or routing jobs for instance. This can also be taken care of using web services. The logic of the mapping system within the web services orchestration is given in Figure 3. This logic corresponds to the 'Mapping System' box in Figure 2. This will be described in section 5. The next section describe what information is involved in the mapping decisions in relation to different areas, in addition to some quality parameters that already exists or could be taken into account within contracts.

Figure 3: The Logic of the Mapping System.



3.1 What decisions need to take place in the mapping

Finding an optimal mapping is a very complex task and has been proven to be NP-Complete[2]. In workflow management systems, in addition to the development of the applications, managing their components and the data flow, the efficient running on the target distributed resources is also a big concern. Recall that there are actually two levels of abstraction, the application level describing the components and the data dependencies, and the execution level which refer to physical instances, either resources or web services. In general, the abstract description does not give any indication on the actual instances for the execution.

In scientific workflows, a range of mapping mechanisms have been proposed where the abstract workflow would undergo a series of transformations or refinements towards optimising the overall performance. These include defining a mapping horison (how far into the future to map tasks as the

[2]In computational complexity, NP-complete refer to a class of problem with no fast solution known (NP stands for nondeterministic polynomial time). In general, heuristics need to be used in order to search towards a solution.

resources / services may change and then the mapping has to be adjusted), many data placement techniques, resources characteristics (load, storage, software / libraries, etc), reducing large workflows techniques (by removing redundant computations or tasks clustering for instance), among many other mapping and scheduling strategies. Some of these techniques are presented in (7), (8), (9).

In business processes mapping, the transactions are based on business contracts which are essentially services agreements. These can be negotiated. SLAs can capture the understanding and the agreed relationship about what the service provider is promising and what the user is expecting and/or interested in. An important matter here is the efficient execution management of the contracts in terms of business, i.e. how to maximise the users satisfaction metrics and minimise penalties for violations for instance (5), (6). In the following section, we discuss some of the quality attributes that would be of interest as part of localisation services description.

3.2 Quality assurance parameters

The quality of service is a substantial aspect for differentiating between similar web services. The agreement / contract would then provide an answer to the following question; which one is the most suitable in terms of cost, time, accessibility, etc., and how to make sure that the provided guarantees with regard to the service are still valid, i.e reviewing the current status of the services. A mapping mechanism is a necessary step in assisting the user picking up the right services with regard to a list of requirements or parameters. In addition to the static attributes like the name, service type, address, or the metadata or standards in use (in relation to the linguistic assets for instance), there are a set of dynamic attributes, resolved at run time (depending on how the services are actually performing). These may include:

- Execution time averages, and total duration
- Cost of the service
- Reliability, availability, or any other characterisation
- Ratings or service level indicators related to the quality of relevant linguistic assets (TM/MTs, glossaries, etc)

- Violation rate
- Levels of user differentiation
- List of Partners, etc.

This is not a comprehensive list, many other attributes can be added in relation to the purpose of the service, the objectives, or the restrictions that may apply. These are not only exposed values from the provider side, but are also generated on the user side. In the next section, we describe a mechanism that is able to gradually take into account predefined users requirements and quality parameters in order to efficiently map a localisation workflow to the set of available web service instances in the Cloud.

4 Mapping mechanism

The following lines describe the steps involved in the selection of the service instances. The proposed solution aims at achieving a flexible mechanism able at gradually taking into account predefined requirements and quality parameters to efficiently map an abstract localisation workflow to the set of available service instances in the Cloud. The solution provided is obviously not unique. Here are the steps of the mapping algorithm:

- Calculation of initial attributes ratios in relation to the cost and execution time, which are the piloting parameters. These ratios represents how valuable is a service for a given attribute with regard to its cost and the execution time (the execution time takes into account the response time of the service):
  - For every attribute and instance, calculate the cost ratio and time ratio. For Accessibility for instance: $CostRatio_{ij} = \frac{Cost_i}{Accesibility_i}$ and $TimeRatio_{ij} = \frac{Exec\_Time_i}{Accesibility_i}$, where index $_i$ represents the service instance and index $_j$ the attribute (Accessibility in this case).
- The initial mapping is done with regard to the user defined requirements and thresholds, i.e the minimum required values of the attributes specified by the user without violating the overall time and cost constraints. Otherwise, for each service type, the service instance that meet the minimum cost is selected. If either of the initial mapping

cases violate the overall cost, the workflow cannot be mapped based on the user request and the algorithm ends.

- Calculate the initial sum of ratios for each attribute and selected service instance, i.e $\forall j \in A$, $\forall s \in S_{initial}$, $InitialSum_{sj} = CostRatio_{sj} + TimeRatio_{sj}$ where A is the list of attributes and $S_{initial}$ is the initial instances selection.

- For each attribute:

  - find the most valuable instances based on the available attributes ratios. The user can specify a list of parameters that he/she want to be considered (availability, reliability, ...). Otherwise, all available parameters are taken into account. All services with lower sum of ratios are selected: $\forall i \in S$, $\forall j \in A$, $CostRatio_{ij} + TimeRatio_{ij} < InitialSum_{sj}$.
  - Calculate the actual differences in the attributes values.
  - Calculate the ratios of the differences, sort the related instances, and select the service instance with the lowest sum of ratios. This ensures that the instance with best value is selected from the list, i.e. it is preferable to spend 4 more time units looking to increase the accessibility by 5% than 7 time units for 6% increase.
  - Ensure that the new selection of service instances meets the user defined cost and time requirements. Otherwise, select the next instance in the sorted list.

- Finally, ask the user to rate the selected services in order to add it to the list of parameters. This is intended to generate a rating ratio and add it as a potential selection parameter.

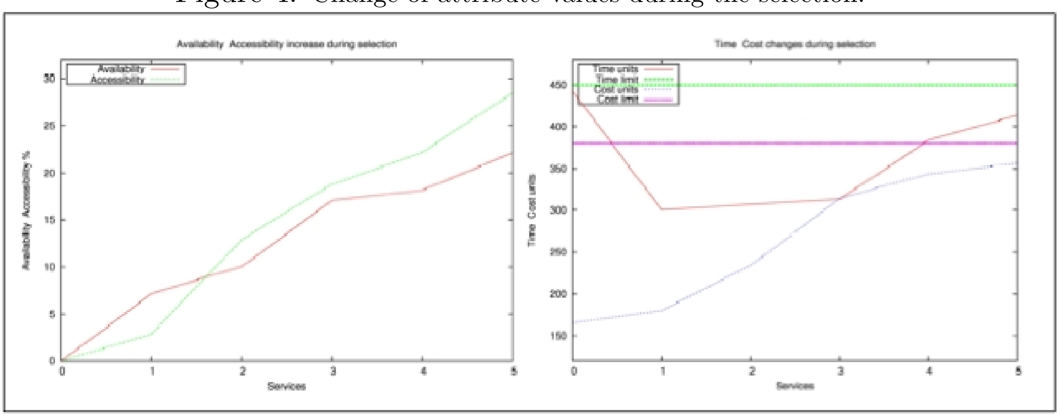
4.1 Simulation & Discussion

The next lines describe the interface that is planned to support the mapping based on the logic presented in Figure 3, in addition to a small simulation run. The user starts by describing a new localisation process using a high-level composition representation. From this composition, the list of the available web services is automatically set up using services catalogs and discovery.

The system will check access to these services which may require a subscription for instance. SLAs generation could be done at this level. Initial sites selection or level grouping can then be carried out before the actual mapping. The user provides constraints and requirements for the QoS parameters using a user interface that shows the list of exposed parameters from the service providers, and locally built indicators from the user's perspective (using the logs and ratings).

The mapping mechanism is then applied and the composition using a process modeling language is generated for execution. An example of a mapping on a workflow with 5 service types and about a dozen instances per service type is given in Figure 4. In this example, the user sets the overall cost and time thresholds, and two attributes were taken into account; availability and accessibility. In comparison to the initial mapping, the mechanism has gradually increased the overall availability and accessibility throughout the selection process (by more than 22% and 28% respectivelly) without violating the cost and time requirements. This shows the importance of such a mechanism even with a relatively small set of instances per service type.

Figure 4: Change of attribute values during the selection.



One of the most important challenges, which will need a substantial effort in localisation web services is the interoperability, not only in the discovery and services intercommunication (as already mentioned), but more importantly from the metadata perspective. This will obviously have an impact on the mapping. The services are developed and deployed by various sources, and might use different localisation / translation standards and various notations. Standardised and publicly published localisation web services are needed. In this aspect, introducing semantic constraints or objects will help the user in validating the components (templates or components represented in ontologies for instance), which will include information about the meta-data. Augmenting the abstract level with semantic descriptions will also enables the search of previous instances or templates in the case of similar processes, and reducing the cost of developing new workflows and then their mapping. For the mapping mechanism, we will also investigate the possibility to allow more than one instance per service type (similar to parallel batching). This will allow to decrease the execution time for very large or time critical tasks.

5 Conclusion

Automated natural language processing techniques and services are becoming increasingly viable. Today, these advances in technologies, along with more efficient computational and infrastructural support, can play a very important role in addressing the critical issue of information imbalance. Indeed, the efficient IT consumption and service delivery in Cloud computing, coupled with web services, can efficiently tackle the huge volume of content with high flexibility and acceptable quality for a large range of language pairs.

In this paper, we have discussed the use of a service-oriented approach in the development of translation and localisation workflows, and specially how to efficiently map abstract workflows into actual instances. The maturity of process modeling languages and standards are key aspects to the support of efficient localisation service-oriented architectures. However, the lack of standard interfaces and terminologies are still need a substantial investigation effort. Our aim was primarily to introduce a way in selecting the most relevant service instances per service type. It provides a dynamic support in service reviewing and organisation which is a necessary support to service-oriented localisation and translation.

Acknowledgment

This research is supported by Science Foundation Ireland (Grant 07/CE/I1-142) as part of the Centre of Next Generation Localisation (www.cngl.ie) at the University of Limerick.

References

[1] *Centre of Next Generation Localisation*. http://www.cngl.ie

[2] *Web Services Essentials*. E. Cerami. O'Reilly Media.

[3] *Unraveling the Web Services: An Introduction to SOAP, WSDL, and UDDI*. F. Curbera et al. IEEE Internet Computing, Vol. 6, No. 2, 2002.

[4] *Language Grid Project*. http://langrid.nict.go.jp/en/index.-html

[5] *Analysis on Service Level Agreement of Web Services*. HP Labs, 2002.

[6] *Foundations of Service Level Management*. R. Sturm. SAMS, 2000.

[7] *Pegasus: A framework for mapping complex scientific workflows onto distributed systems*. E. Deelman et al. Scientific Programming, Vol. 13, No. 3, 2005.

[8] *Persistent Workflow on the Grid*. L. Aouad et al. The 2008 IEEE Asia-Pacic Services Computing Conference.

[9] *Data Management Challenges of Data-Intensive Scientific Workflows*. E. Deelman and A. Chervenak. 2008, 8th IEEE International Symposium on Cluster Computing and the Grid.