Localisation Focus

THE INTERNATIONAL JOURNAL OF LOCALISATION

ISSN 1649-2358

Centre for Next Generation Localisation

Fondúireacht Eolaíochta Éireann
Science Foundation Ireland

The peer-reviewed and indexed localisation journal

VOL. 10 Issue 1

Localisation Focus
The International Journal of Localisation
VOL. 10 Issue 1 (2011)

CONTENTS

Editorial
Reinhard Schäler 3

Research articles:

An Argument for Business Process Management in Localisation
David Filip, Eoin Ó Conchúir…….....4

Enabling Complex Asian Scripts on Mobile Devices
Waqar Ahmad, Sarmad Hussain…….....18

LocConnect: Orchestrating Interoperability in a Service-oriented Localisation Architecture
Asanka Wasala, Ian O'Keeffe, Reinhard Schäler…….....29

Localisation in International Large-scale Assessments of Competencies: Challenges and Solutions
Britta Upsing, Gabriele Gissler, Frank Goldhammer, Heiko Rölke, Andrea Ferrari …….....44

EDITORIAL BOARD

PUBLISHER INFORMATION

Editor: Reinhard Schäler, *Director*, Localisation Research Centre, University of Limerick, Limerick, Ireland
Production Editor: Karl Kelly, *Manager* Localisation Research Centre, University of Limerick, Limerick, Ireland
Published by: Localisation Research Centre, CSIS Department, University of Limerick, Limerick, Ireland

AIMS AND SCOPE

Localisation Focus – The International Journal of Localisation provides a forum for localisation professionals and researchers to discuss and present their localisation-related work, covering all aspects of this multi-disciplinary field, including software engineering, tools and technology development, cultural aspects, translation studies, project management, workflow and process automation, education and training, and details of new developments in the localisation industry. Proposed contributions are peer-reviewed thereby ensuring a high standard of published material. Localisation Focus is distributed worldwide to libraries and localisation professionals, including engineers, managers, trainers, linguists, researchers and students. Indexed on a number of databases, this journal affords contributors increased recognition for their work. Localisation-related papers, articles, reviews, perspectives, insights and correspondence are all welcome.

Subscibers to the print edition of Localisation Focus - The international journal of Localisation can access an archive of past issues online.

Subscription: To subscribe to Localisation Focus - The International Journal of Localisation www.localisation.ie/lf

Localisation Focus – The International Journal of Localisation (ISSN 1649-2358) is published and distributed annually and has been published since 1996 by the Localisation Research Centre, University of Limerick, Limerick, Ireland. Articles are peer reviewed and indexed by major scientific research services.

FROM THE EDITOR

Localisation is now firmly established as an academic discipline and part of the academic canon. It is time to take stock, to look back over the 16 years of work of the Localisation Research Centre at the University of Limerick, the large body of academic publications now available in our discipline, and to venture a view into the future. Social Localisation, driven by users rather than enterprises, will certainly become a defining part of this future. Mobile devices and languages not known in mainstream localisation today will require a radical change in the way we perceive localisation. Mapping out and understanding the processes underlying these changes will become paramount.

In their contribution, David Filip and Eoin Ó Conchúir present a strong argument for the use of Business Project Management in Localisation. They present three case studies to illustrate how BPM can help us to understand and meaningfully react to the constantly evolving state of localisation and the emerging and powerful evolution of user-driven localisation. The use cases cover the content authoring business logic of WordPress, the traditional localisation process used by large, medium and small enterprises, and the localisation process deployed by nonprofit businesses.

The explosive growth of wireless networks and mobile devices in emerging markets and developing regions of the world have opened up new avenues for localisation. More than ever before, localisers need to understand the specific challenges and problems associated with mobile device localisation – and, specifically, those requiring the enabling of complex Asian scripts. Waqar Ahmad and Sarmad Hussain highlight the need for making mobile devices accessible in the local languages (and scripts) of the growing user population in Asia and in domains as diverse as education, health, entertainment, business, sports, and social networks. Their contribution, Enabling Complex Asian Scripts on Mobile Devices, reports on the successful deployment of an open source rendering engine, Pango, on the Symbian platform for Urdu, Hindi, and Khmer.

Interoperability is one of the areas in localisation research that probably attracted most attention in 2011, especially in the context of the increased traction of the XML-based Localisation Interchange File Format, XLIFF, among both academic and industrial researchers, as indicated by the highly successful and now well-established series of XLIFF Symposia. Asanka Wasala, Ian O'Keeffe, and Reinhard Schäler report on Orchestrating Interoperability in a Service-oriented Localisation Architecture using LocConnect within a service-oriented architecture (SOA) framework.

A team from the German Institute for International Educational Research and cApStAn Linguistic Quality Control cover an area of research that has been largely unreported in the literature and at localisation events, namely the challenges encountered, and the solutions provided by researchers and practitioners working on the localisation of International Large-scale Assessments of Competencies. Britta Upsing, Gabriele Gissler, Frank Goldhammer, Heiko Rölke, and Andrea Ferrari take the Programme for International Student Assessment (PISA) and the Programme for the Assessment of Adult Competencies (PIACC) as an example and describe how their groups dealt with the specific challenges in this brand-new area of internationalisation and localisation.

In 2012, this journal will expand its reach in Africa reporting on the significant localisation activities taking place on this exciting continent. In addition, we will work on a thorough survey of research in localisation, providing easy access to the body of work now available.

Finally, on behalf of the editorial team, I would like to thank the Centre for Next Generation Localisation (CNGL) for its generous support, and the more than 20 international members of our editorial board for their continued and enthusiastic assistance to develop and grow Localisation Focus – The International Journal of Localisation, the world's first peer-reviewed and indexed academic journal in localisation.

Reinhard Schäler

An Argument for Business Process Management in Localisation

Dr. David Filip, Dr. Eoin Ó Conchúir
Centre for Next Generation Localisation,
Localisation Research Centre,
CSIS Department,
University of Limerick,
Ireland
www.cngl.ie
www.localisation.ie
davidf@ul.ie, eoin.oconchuir@ul.ie

Abstract

Enterprise-level translation management systems cater well for their well-defined use cases. With the rise of user-generated content, the world of localisation is extending to include what we term as 'self-service' localisation. The localisation needs of this emerging environment may differ from more traditional enterprise-level scenarios. In this paper, we present an argument for using business process management (BPM) to help us better understand and define this emerging aspect of localisation, and we explore the implications of this for the localisation industry. Modelling a business process allows for that process to be managed and re-engineered, and the changes in efficiency quantified. It also helps to ensure that automated process aids and electronic systems are put in place to support the underlying business process, matching the real needs of its stakeholders. In this paper, we specifically look at emerging self-service localisation scenarios in the context both of the evolution of the traditional industry process as well as in the context of not-for-profit localisation.

Keywords: *: business process management, BPM, modelling, user-generated content, self-service localisation*

1. Acronyms Used and Basic Definitions[1]

BI - Business Intelligence. The process and technology of organising and presenting business process data and meta data to human analysts and decision makers to facilitate critical business information retrieval.

Bitext - a structured (usually mark up language based) artefact that contains aligned source (natural language) and target (natural language) sentences. We consider Bitext to be ordered by default (such as in an XLIFF file - defined below, an "unclean" rich text format (RTF) file, or a proprietary database representation). Nevertheless, unordered Bitext artefacts like translation memories (TMs) or terminology bases (TBs) can be considered special cases of Bitext or Bitext aggregates, since the only purpose of a TM as an unordered Bitext is to enrich ordered Bitext, either directly or through training a Machine Translation engine.

Bitext Management - a group of processes that consist of high level manipulation of ordered and/or unordered Bitext artefacts. Usually the end purpose of Bitext Management is to create target (natural language) content from source (natural language) content, typically via other enriching Bitext Transforms, so that Bitext Management Processes are usually enclosed within a bracket of source content extraction and target content re-importation.

Bitext Transformation - Similar to Bitext Management, but the Bitext is enriched with newly created or manually modified target content. The agents in Bitext Transformation may be both man and machine, or any combination of the two.

BOM* - Bill of Materials

BPM - Business Process Management

CAT* - Computer Aided Translation

[1]For standard localisation industry acronyms see MultiLingual 2011 Resource Directory (MultilLingual 2011). Such standard industry terms are marked with an asterisk (*). We also give short definitions for terms that may be considered commonplace to prevent misunderstanding.

ESB - Enterprise Service Bus, an open standards, message-based, distributed integration infrastructure that provides routing, invocation and mediation services to facilitate the interactions of disparate distributed applications and services in a secure and reliable manner (Menge 2007).

HB - Hand Back. This is being used systematically in two related meanings, either as the message/material conformant to a related HO BOM, leaving an organisation/swimlane as response to the HO, or the last process/subprocess that happens before the corresponding pool-crossing flow.

HO - Hand Off. This is being used systematically in two related meanings, either as the message/material leaving an organisation/swimlane to solicit a response conformant with its BOM, or the last process/sub-process that happens before the corresponding pool-crossing flow.

IS - Information System

LSP* - Language Service Provider

Man - used as synonymous with human, not male, such as for 'man-hours'.

Message - the token in an ESB facilitated workflow or generally any SOA driven workflow. Messages are being enriched as they travel through workflows.

MLV* - Multilanguage Vendor, a type of LSP.

NFP - Not-for-profit

Process - procedure consisting of logically connected steps with predefined inputs and outputs.

SLV* - Single Language Vendor, a type of LSP.

SMB* - small and medium-sized businesses

SOA - Service Oriented Architecture, an architecture concept which defines that applications provide their business functionality in the form of reusable services (Menge 2007).

Swimlane - Pool and Lane as used in BPMN not in sports.

TM* - Translation Memory

TMS* - Translation Management System

Token - whatever travels through a defined process or workflow. Each token instantiates the process or workflow. In this sense, multiple instances of a workflow are created not only as different tokens entering the predefined processing but also at any pre-defined point in the workflow or process where tokens are split according to business rules.

Workflow - an automated process. This is not a commonplace distinction, but we coin it for practical convenience.

XLIFF* - OASIS XLIFF, i.e. XML Localization Interchange File Format. We mention XLIFF in its capacity as a token in localisation processes and as a message being enriched in an ESB or SOA based workflow.

XOR - exclusive OR, logical connective. Used here to characterise the exclusive gate in modelling, as used in BPMN (2011).

2. Introduction

In its essence, localisation is driven by users' preferences to access information in their native language, and this is no different for information being presented online (Yunker 2003). In the corporate context, this has lead to companies providing localised versions of their websites, for example (Jiménez-Crespo 2010).

Meanwhile, with the widespread availability of 'Web 2.0' platforms, it is not only corporations themselves that are producing localisable and localised content (O'Hagan 2009). For example, fans of certain publications (in this case, comics) have produced unsolicited user-generated translations in a collaborative manner (O'Hagan 2009). Indeed, user-generated content (be it opinions or otherwise) is nothing new, although the possibility to work collaboratively online is relatively new. The involvement of online communities in translation has evolved to become solicited user-generated translations. This general concept of leveraging the latent talent of the crowd, particularly online, was coined as crowdsourcing (Howe 2006).

The shift in how content is being transformed in the localisation and translation world has been termed the "technological turn" (Cronin 2010). With respect to content distribution, Cronin argues that the most notable change has come in the form of electronic work station PCs being gradually replaced by the use

of distributed mobile computing. This transition is leading to Internet-capable devices becoming ubiquitous. Rather than localisation being driven by the need to produce static centrally-created content, the emergence of user-generated content is leading to the localisation of user-generated content into personalised, user-driven content. Internet-connected platforms present the potential of collaborative, community translations. This is in contrast to the commercial option of translation through employed translators, freelance translators, or the use of a localisation vendor to act as an intermediary.

While enterprise-based localisation of content and software, being produced in-house, is a mature process with quality assurance certifications available (Cronin 2010), the involvement of online communities (or the "crowd") in localisation is a relatively newer field. Similar to the concept of "open sourcing", the crowdsourcing of localisation is outsourcing the tasks involved to an "unknown workforce" (Ågerfalk and Fitzgerald 2008). We assume that in such a context, contractual agreements may not be in place with members of the community. Rather than being able to agree binding deadlines with paid translators, community members may offer to work on translation tasks on a whim (depending on the process put in place).

In this paper we argue that the evolved state of localisation is yet to be fully understood. Indeed, there is a constant evolution of how the concept of user-driven translation can be applied in real-world situations.

In the following sections, we argue that the activity of business process management (BPM) is a valuable tool for allowing us to understand the new requirements of information systems involving user-generated content and user-provided translations. In later sections, we present three case studies to illustrate how BPM may be applied, and what may happen if the underlying business processes are not correctly incorporated into a new information system. Finally, we conclude that given the advancement of self-service localisation, even in the corporate context, such emergent business processes can be better addressed through BPM.

3. New Business Processes, and Business Process Management

On the subject of newly-emerging business processes in localisation, we must define how a certain block of content to be localised will be ultimately used. To illustrate this point, let us compare the difference in expectations between the localised version of a corporate brochure when contrasted with that same corporation's desire to localise its ongoing social media stream for different locales. With the former example, we may expect very formal and accurate language, whereas the latter may allow for a more informal approach. A further distinction may be made between relatively informal content being produced by a corporation and useful customer-generated content that may benefit other customers of different native languages. An example of this would be a descriptive forum message, posted online by a customer, providing a solution to an issue with a company's product. Indeed, translation quality is a multidimensional concept that can be approached in different ways including process-oriented international standards, or more community-based localisation (Jiménez-Crespo 2010).

To illustrate that point, we present Table 1. The table shows how content coming from different sources may be localised using different approaches. The upper-left quadrant may be seen as the traditional route taken in localisation. Such business processes are the main focus of translation management systems. The upper-right quadrant may be too costly compared to the value it produces, since a constant stream of user-generated content may overwhelm traditional localisation processes. Indeed, companies are presented with the emerging choice of facilitating their online community in localising content that has been produced by their peers. The lower two quadrants are of particular interest, as it is here that a community of translators (the "unknown workforce") may be asked to help with the localisation of content. It should be noted that volunteer translators are not necessarily individuals donating their free time, but also representatives of external organisations who would benefit from having the content made

	Traditionally-generated content	**User-generated content**
Traditional content localisation	Localisation of corporate-controlled content by a paid contracted entity (such as a localisation service provider).	Localisation of user-generated content by a paid contracted entity (such as a localisation service provider).
User-driven content localisation	Localisation of corporate-controlled content by volunteer community members.	Localisation of user-generated content by volunteer community members.

Table 1: Both in-house and community-generated content may be localised by either commercial localisation vendors or by the community itself.

available in their primary language.

Focusing on any of these four quadrants in Table 1 presents us with different business processes being represented. For example, a system allowing for ad-hoc volunteer translations of short social media messages may have quite different requirements to a system involving tightly-controlled contracted freelance translators. In the following sub-section, we argue that it is critical that the underlying business processes be closely matched by the functionality of electronic systems designed to support them. We explain how a mismatch in information technology (IT) strategy with information systems (IS) strategy along with business strategy may lead to practical failure of the system being produced.

3.1 Information Systems Perspective

In the localisation context, a "system" may be the socio-technical entity that supports traditional enterprise-based localisation, or a user-driven localisation scenario. To discuss how systems may be designed to cater for any particular permutation of the localisation process, we must first address the nature of a system itself. In information systems theory, the "system" does not merely refer to a computing machine such as a personal computer (PC). Neither does it refer simply to a software application (large or small, TMS, ESB etc.) designed to facilitate certain operations. Rather, we view an information system as a socio-technical entity, similar to Galliers (2004).

An information system is comprised of the information being processed and produced, along with the organisational context of its users and other stakeholders. An information system designed to encompass a socio-technical environment would combine information and knowledge sharing services that would facilitate both the exploration and exploitation of knowledge (Galliers 2006).

A long-standing view of information systems is that the activities falling under information communications technology (ICT) development must be closely aligned to the information system as a whole, which in turn must be aligned to the organisation's business strategy (Galliers 2006). A misalignment between these concepts or activities may lead to a failed system. A failure does not necessarily imply that the system itself does not function (Laudon and Laudon 1996). For example, a system may be perceived as failed if it has not been successfully adopted by its intended user base, even if the system itself runs "as designed". In this paper, technology underlying localisation including CAT tools and Translation Management Systems (TMS) is discussed from this broader IS perspective. As such, they need to be aligned with business objectives.

3.2 Business Process Management (BPM)

A business process is a "set of partially ordered activities intended to reach a goal" (Hammer and Champy 1993). Relating this to localisation, a high-level business process may be taking a mono-lingual technical manual and all the steps required to adapting it to various target locales. Similarly, a business process may describe the activities required to produce a community-based localisation project. In localisation specifically, Lenker et al (2010) argue that by abstracting a localisation business process as a workflow, the process can be potentially automated and its efficiency improved. Business processes may be quite low-level, with a large organisation being comprised of thousands of such processes (Turban et al 1993).

Formally, a process is seeded with inputs, and it produces outputs. Thus, the output of a process can be measured. This is an advantageous approach, since measurements of process efficiency allow us to tweak the process and measure the consequences. BPM thus provides a structured framework for understanding the business process itself, and then optimising that process.

3.3 Modelling Business Processes

An information system may be developed to improve the current workings of an organisational unit, or it may be conceived to support an entirely new set of business activities. In either case, we may analyse the underlying business activities, producing conceptual models of the activities.

Modelling a business process is the act of formally describing the business processes at hand. Many businesses have process models of their systems (Cox et al 2005). Once contextual information has been elicited about the socio-technical system, and explicitly described through business process modelling, an understanding of what problems need to be solved should emerge (Cox et al 2005).

Business processes can be captured in a standard language, that being Business Process Model and Notation (BPMN, formerly also known as Business Process Modeling Notation). It is maintained by the Object Management Group (OMG). It offers an

extensive standard modelling framework, readily understandable by business people, including analysts and technical developers (BPMN 2011). Models recorded in this manner allow for the business processes to be modelled while abstracting from actual implementation details. This provides a standardised way of communicating process information to other business users, process implementers, customers, and suppliers. Requirements engineering approaches can be applied to BPM, such as employing role activity diagrams (Bleistein et al 2005).

By taking a set of models produced in a standard modelling language, BPM can let us carry out business process improvement through business process re-engineering. Software tools allow the analyst to work on the business process models in order to produce an optimised set of processes, ultimately improving the workings of the organisation.

4. Case studies

In this section, we present a number of case studies to demonstrate the concepts behind BPM, and how they may be applied to localisation. These case studies are then compared and contrasted in the following analysis and discussion section.

BPM, in essence, deals with understanding the business processes of an organisation. The concept of an organisation here is a socio-technical grouping of people and systems. In order to manage any business process, it is necessary to understand the participants in the system, the activities taking place in the system, and the message flow of information throughout the system (BPMN 2011). For example, Lewis et al (2009) analyse the set of activities and communication mechanisms involved in a traditional localisation workflow, and use this to understand newer community-based approaches to localisation. First, though, we present a simple example of a system that supports the business logic of content creation.

4.1 Case Study 1: Content authoring business logic encapsulated by WordPress

With the advent of the World Wide Web in the early 1990s, content publishers (both individuals and organisations) were presented with a new opportunity to publish their content. At its most basic, text content can be published online as a hypertext mark-up language (HTML) document by uploading it to a web server. The document can contain static content, and so is limited in how it can encapsulate the business logic of a more complex content system. An information system may be represented somewhat by interlinking static HTML documents. More likely, however, is the need to support the business logic through dynamic server-side scripting which would output HTML documents generated on the fly.

By the late 1990s, a trend in personal web pages was to publish a 'log' of web sites found by the web page owner, in chronological order. Yet, by that stage, most web loggers (who became known as 'bloggers') hand-coded their web sites. No tools were publicly available that would support the requirement of dynamically publishing a series of links to a web page (Blood 2004).

In 1999, a free web logging system called Blogger (http://www.blogger.com) was launched with the tag "Push-button publishing for the people". The simplicity of the system made it very popular, with non-technical users beginning to use the web logging platform simply as a way to publish their thoughts and opinions online, without necessarily any links in the published post (Blood 2004). This was the birth of the blog post format.

At the time of writing this paper, WordPress (http://www.wordpress.org) is one of several popular open-source blogging systems, having first been released in 2003. Perhaps due to the platform's ease

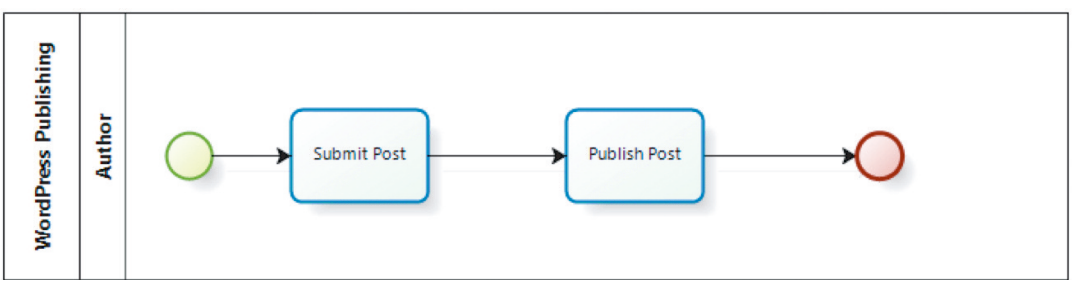

Figure 1: Single-user content authoring and publishing as supported by WordPress.

of use, but moreover its direct addressing of the business logic required by bloggers, the platform has gained a wide user base. WordPress has been adopted by individual bloggers and large organisations alike, such as the popular technology blog TechCrunch (http://www.techcrunch.com) and Forbes' blog network (http://blogs.forbes.com/) (WordPress 2011a).

Figure 1 illustrates the simplest content publishing workflow offered by WordPress. Note that we make use of Business Process Modelling and Notation (BPMN) for the illustrations in this paper. This allows for an abstracted understanding of the underlying business process.

WordPress is a dynamic server-side platform that encapsulates the business process of publishing and managing content online as an individual or as a team of content authors. It does so by supporting the activities of content creation, reviewing, editing, and publishing. WordPress supports the user roles of Super Admin, Administrator, Editor, Author, Contributor and Subscriber (WordPress.org 2011b). A team of content authors may assign these different roles to different people to manage the publishing process. For example, the Contributor role allows that person to author and edit their own content, but not publish it to the blog. An Author user has the same abilities, in addition to being able to publish their own content. Notably, the Editor role can create content, manage their content and others' content, and choose to publish others' content (it is beyond the scope of this article to further describe in detail the roles and capabilities offered by WordPress).

In summary, the system encapsulates the roles and activities required for publishing content online. The business process (the set of activities involved in authoring, editing and publishing online content) is closely matched by the action-centric functionality of the WordPress system. In this case, business process management may be used to understand the underlying business process, to model it, and to tweak it. By illustrating this specific case study of a content management system, we argue more generally that BPM is a worthy approach for understanding the underlying business process, and thus making it more likely that the system being developed will align more closely with actual requirements.

4.2 Case Study 2: The traditional industry localisation process in the industry, enterprise and SMB context

Figure 3 illustrates a high level model of the enterprise localisation process. Each of the high level processes represented by blocks in the figure would need to be defined in further levels of granularity in order to be relevant for real implementations. The model is nevertheless useful as a high-level representation. It is helpful for showing the most important process differences at the relevant level of complexity. In this paper we only include models that can be quickly understood at first glance, for several reasons:

1) To illustrate points made about process differences occurring in different localisation settings.

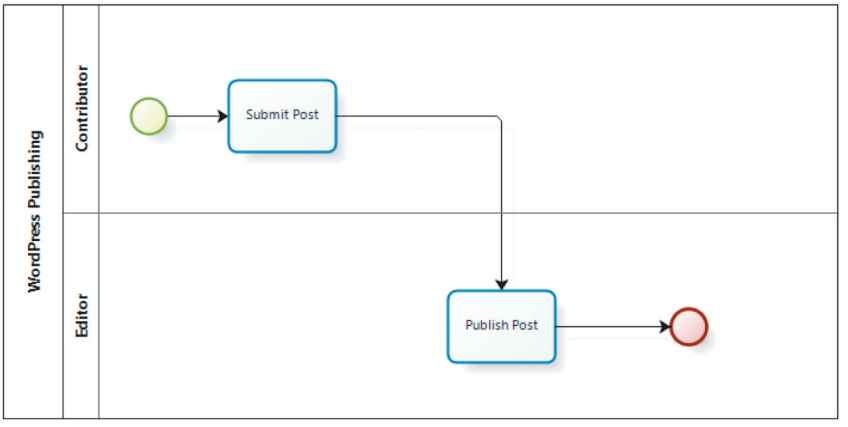

Figure 2: The business process of a Contributor submitting a post, and an Editor publishing that post, as supported by WordPress.

2) To illustrate how the BPMN standard can be used to create pictorial representations facilitating process discussion in a highly intuitive way.

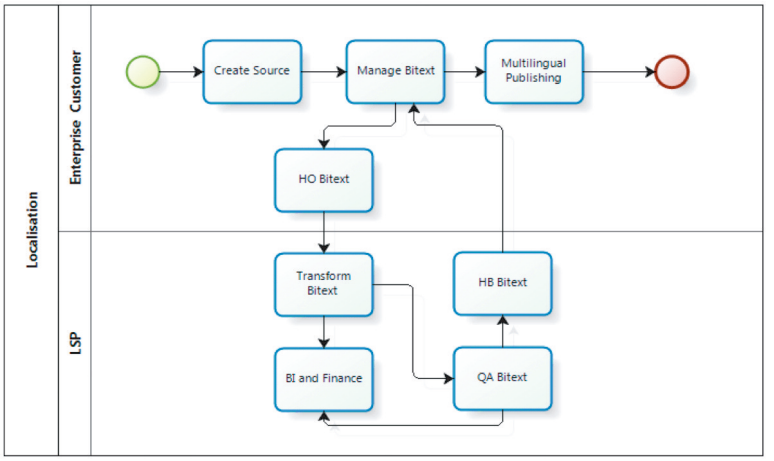

Figure 3: The localisation process in the enterprise context covering content management and publishing.

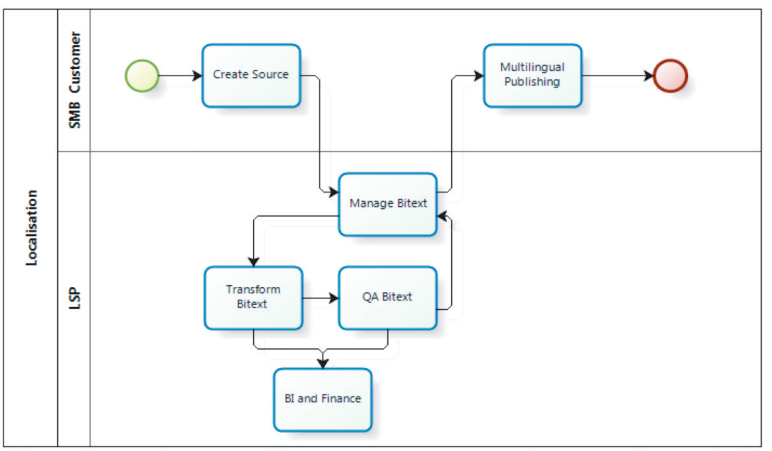

Figure 4: The management of Bitext is usually performed by an LSP partner for an SMB.

The model in Figure 3 anchors the localisation process in the broader context of multilingual content management and publishing. Content is being created specifically in one language, in the sense that a single piece of information can only be conveyed practically in one language at a time. The publisher, however, needs to publish its information in many languages. As the transitions from the creation in one language to multiple languages in publishing always include transformations specific to the language pair, we have labelled the intermediate steps as "Bitext Management". Bitext Management is the central piece of any localisation process. In fact, Bitext Management forms the fundamental distinction between localisation processes in different contexts in terms of whom, where, and how it is executed.

In contrast, Small and Medium Businesses usually lack the resources needed to take control of their translation memory leveraging. They are usually unable to manage their Bitext on their own. Therefore, although localisation customers legally retain rights to their bilingual corpora, in practice their Bitext Management is a black box for them which is managed by a long term LSP partner.

In summary, BPMN has allowed us to visually represent the high-level business processes of Bitext Management for enterprises (Figure 3) and SMBs (Figure 4). It helps to demonstrate that the primary distinction between both cases is whether the "Manage Bitext" activity happens in-house, or is the responsibility of an LSP.

4.3 Case Study 3: The localisation process in the Not-For-Profit context

Further to enterprise and SMB localisation, we would like to investigate whether not-for-profit (NFP) localisation is any different. At a first glance it may seem so. Again, we make use of BPMN to help answer this question.

Figure 5 illustrates a typical localisation process for a not-for-profit customer. It makes use of a low tech SLV, freelance or volunteer translators. While the source content is produced in-house by the NFP organisation, the translation process is performed externally (represented by the "Low tech translate" activity in the figure). "Low tech" is used here in the sense that this scenario does not make any explicit use of Bitext properties, due to an apparent, or real, lack of CAT tools in the process. In particular, the low tech SLV may be an over-the-street agency that only accepts content by fax, sends the content by fax to the translator who types a new document without using translation tools, and the hard copy of the translated document can be rubber-stamped (at a fee) as being translated correctly and accurately by a court-approved interpreter.

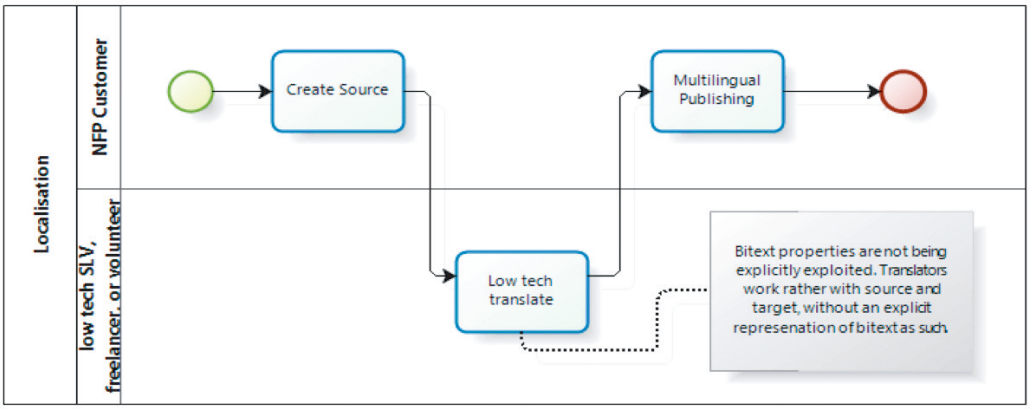

Figure 5: Modelling the localisation process in a not-for-profit scenario.

More generally, this is the low tech scenario of the localisation process typical for low Localization Maturity Levels (DePalma 2006; DePalma 2011; Paulk et al 1993). The business process is not specific to not-for-profit organisations. This has important implications for those building localisation solutions for not-for-profits that may have fewer resources in place to support the localisation process. Such service and technology solutions would need to address a certain level of effectiveness, and hence sophistication. As a result, the solutions would need to take responsibility for Bitext Management, as the typical NFP customer will not be able to manage their Bitext on their own. Organisations that are aiming to

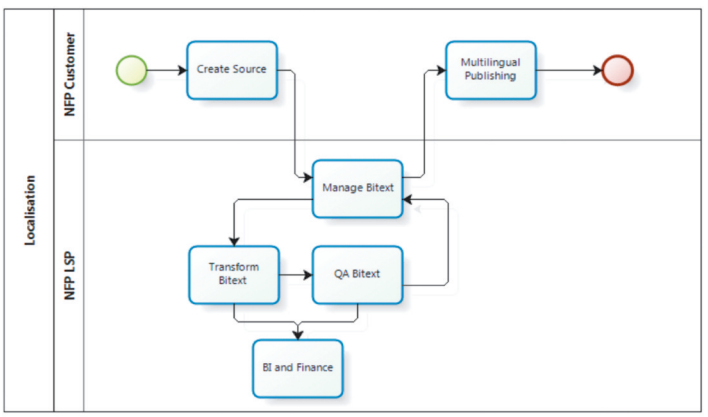

Figure 6: The localisation process in the not-for-profit context features Bitext Management outside of the organisation.

[2] CSA gave a preview of the 2011 TMS report on 8th September, 2011. However, the full report was still pending publication at time of writing.

support not-for-profit localisation may - in effect - emulate the SMB localisation model, at least at this high structural level. Figure 6 illustrates this finding.

One may therefore come to the conclusion that there is no difference between the traditional localisation process (Figure 4) and the not-for-profit model (Figure 6). However, in section 5.3 we describe why this is actually not the case.

5. Case Study Analyses

In the previous sections, we presented three case studies by modelling the relevant business processes. Some comparisons were made between the case studies. In this section, we discuss how the existing localisation solutions address the above described scenarios and present further conclusions arising from the analysis of these case studies.

Localisation platforms, such as CAT tools and Translation Management Systems (TMS), do currently exist and primarily address the traditional enterprise localisation process. We wish to understand the level and nature of impact of next generation localisation factors that we see arising with the inclusion of crowd sourcing concepts. To do so, we need to discuss the role of CAT tools and TMSs in the localisation-enabling Information Systems (IS).

5.1 The role of current platforms in addressing localisation business needs

Since 2006, Common Sense Advisory (CSA) has been publishing an authoritative comparison of translation management systems (TMSs) (Sargent and DePalma 2007 and 2008). As there has not been a comprehensive report since 2008 (only individual TMS scorecard additions have been published)[2], the 2008 report still serves to define classifications and groupings. Our classification in this paper draws loosely from the CSA classification.

The most prestigious category according to CSA is the Enterprise TMS (ETMS) or "cradle to grave" systems. These systems are expected to be enterprise-class information and automation systems. Many players have been trying their luck in this category. The initiator and long time leader of this category had been Idiom WorldServer (now SDL WorldServer), which, even today, remains unparalleled in the expressivity of its workflow engine within the class of ETMSs. However, this class of TMSs is being rendered largely obsolete due to the present-day development of general enterprise architecture, in terms of business need and development.

It has been noted (Sargent and DePalma 2008; Morera et al 2011) that localisation automation systems have been successful in narrowing permissible workflow complexity in building a particular production workflow. Complexity here refers, roughly, to the number of the classical workflow patterns (van der Aalst et al 2003; Morera et al 2011).

TMSs can be considered as highly specific automation systems, and different categories of TMSs may be distinguished by their level of specificity for localisation workflow support. Part of their success is in simplification relative to traditional industry patterns.

For instance, most of the existing systems are hard wired for a single source language per project. This means that they will be challenged by multiple source languages scenarios that play an increasingly important role. The reason that current solutions have been built to cater exclusively for a single source language scenario is that most of the current enterprise-class localisation processes actually normalise to a single source language, very often English, especially in multinationals. Even Asian and German-based multinationals, that would often try to use their local languages as the source languages, are forced to use English due to outside forces. Such forces would include the present state of the market and procurement necessities such as economies of scale. If English is not used as a source language, it still tends to be used as a pivot language, through which all content is translated. In the following, however, we leave aside the complexities of managing multiple source languages.

The least capable, in terms of building complex automation workflows, would be the category of TM Servers. The capabilities of TM Servers in the area of automation can range from a simple automated segment pair lifecycle through to a predefined set of states that each pair can retain throughout its life, all the way from 'new', through to 'revised' and to 'deprecated'. Every product in this category manages to automatically search and retrieve relevant terminology, both for full and fuzzy matches.

However, this capability has been commonplace in our industry for so long that it is not even considered "automation". It is, indeed, a level of automation that

can be taken for granted thanks to the native functionality of computer aided translation (CAT) technology and is usually not enhanced to a great degree by server-level products (apart from the apparent advantages of committing to a regularly backed up well-resourced database, compared to a locally installed database or a local proprietary database file).

In fact, many tools that had been working without issue locally or through local area networks (LAN) had maturity challenges when introducing or perfecting their server-based product. The leader in this capability has, so far, been the Lionbridge Translation Workspace that is offered through the GeoWorkz.com portal (originally known as Logoport).

We see a tension between the interests of large LSPs in attempting to control the technology space, while customers seek to avoid technology lock-in. There are repercussions of this tension for the LSP world. An LSP may have a significant number of stakeholders. Various types of LSPs exist ranging from mutually-coordinated freelancers, to bricks-and mortar SLVs, through to large multimillion so-called MLVs competing for a place on the CSA beauty contest ladder (Kelly and Stewart 2011).

The standardisation driven by enterprises will be exploited downwards and we expect that this will lead to the language industry becoming even more strategic, yet even more commoditised. We predict that there will be no differentiator for SLVs except for resource management. MLV competition will become even fiercer as the standardised SOA and ESB based architecture will drive the cost of entry even lower. Cyclically, the MLVs will need to deal with large enterprises taking Bitext Management and other value added high margin services in house, forming specialised service units such as Oracle's Ireland based WPTG (Worldwide Product Translation Group**).**

5.2 Adoption of Crowdsourcing in Localisation

The democratisation of the Web has emerged through the power of the "crowd". This trend has also been increasingly applied to the localisation process where the concept of crowdsourcing has seen members of the crowd performing localisation tasks, such as translation and reviewing. There are two settings in which the stakeholders are ahead in embracing this relatively new trend:

1) Enterprises

2) Not-for-profit (NFP)

The crowd is important for both of these because of similar, yet distinct, reasons. In the not-for-profit (and potentially charitable) setting, accessing a crowd of volunteers would be attractive. Crowd-sourced translation may also be attractive for enterprise, but there are significant levels of investment required for supporting that through technology, oversight and management. In other words, the return on investment (ROI) must still be properly calculated even if engaging with an unpaid crowd.

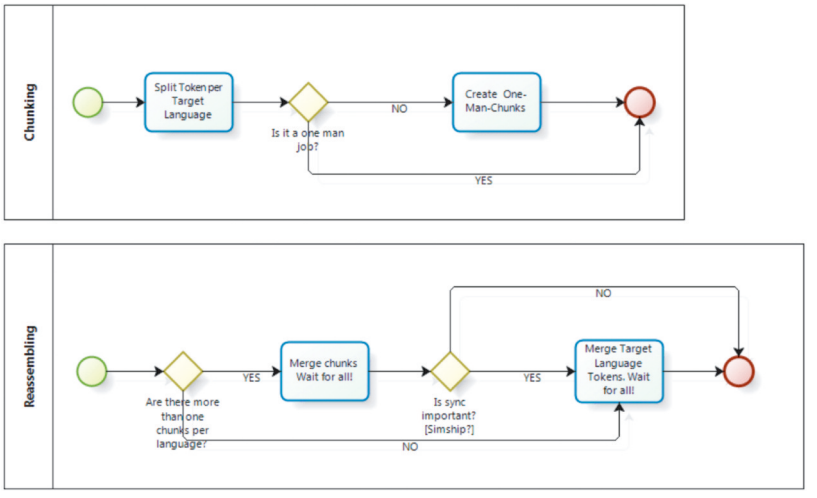

Figure 7: The chunking and reassembling activities in a typical localisation process.

We speculate that the motivation of the unpaid crowd may be a distinguishing factor in next generation localisation. This may not be such an issue in a more traditional paid translation context.

More specifically, volunteers may have little time to contribute to a localisation project. The implication of this is profound: the chunks of content presented to them as tasks need to be much smaller than those required in the traditional localisation workflow. We discuss this topic further in the next sub-section.

5.3 New Requirements for Bitext Chunking

Figure 7 shows the lower level models of chunking and reassembling that we have been using in previous models when referring to Bitext Management.

The chunking process multiplies the tokens that are travelling through the process in two steps. First, it creates a token per target language. Second, it creates a token per one-man-chunk.

A process that uses chunking must also contain reassembling further down the road to ensure that tokens are properly merged back (i.e. well handled). One may notice that the re-merging of target versions into one deliverable token is optional and more likely to occur in an industry setting than in a not-for-profit setting.

Using XLIFF as the message container provides benefits as XLIFF is capable of carrying a token in the size of thousands of files, or as small as a single translation unit (OASIS XLIFF 2008).

Figure 8 applies equally to the industry setting and the not-for-profit setting. There is, however, a very important parameter that governs the behaviour of the XOR gateway diagram. From a technical perspective, the decision is simply based on a single parameter.

Figure 8 represents the process of abstracting the steps that are needed to be taken to get a certain output, given an input. The figure does not itself specify whether or not the workflow process needs to be automated in real life. The parameter is the size of a one-man-chunk. In the paid industry setting the one-man-chunk may easily comprise effort of up to five man-days (in case of relaxed schedules even ten man-days may count as one-man-chunks, and in the literary translations world one person routinely deals with effort in terms of man-months).

However not-for-profit organisations may have to deal with real life emergencies as they arise (such as tsunamis, earthquakes, famines, and many other less dramatic, yet time sensitive, issues). Therefore, they may have very tight schedules as in the translation industry, but seldom have the budgets to buy full-time resources.

Therefore, the one-man-chunk in the volunteering setting is better defined in terms of man-hours. The five-man-day chunk is not extraordinary for enterprise settings, but could take months for a volunteer to complete. As such, the content requires a much higher level granularity of chunking for fast turnaround of each chunk.

Assuming that a not-for-profit project needs to publish multilingual information within a week of the creation of the source text, and assuming that the crowd of highly-motivated volunteers have on average 20% of normal full-time employment to dedicate to the project, we conclude that a project should be chunked accordingly to blocks of four man-hours.

In the case of more stringent deadlines, or where the crowd is less disciplined, chunking may need to be set at two man-hours, or smaller.

Chunks smaller than one man-hour may not be effective in practice, unless the tasks are specialised, such as for user interface translation projects.
Following this discussion, we can see the typical model for NFP localisation should be as illustrated in

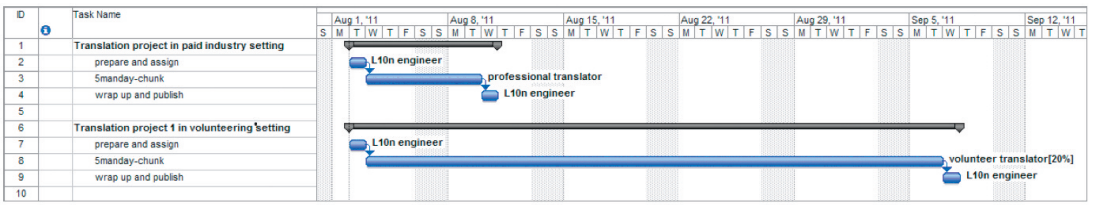

Figure 8: Industry chunking is not for volunteers

[3] See classic discussion of workflow expressivity by Aalst et al. 2003.

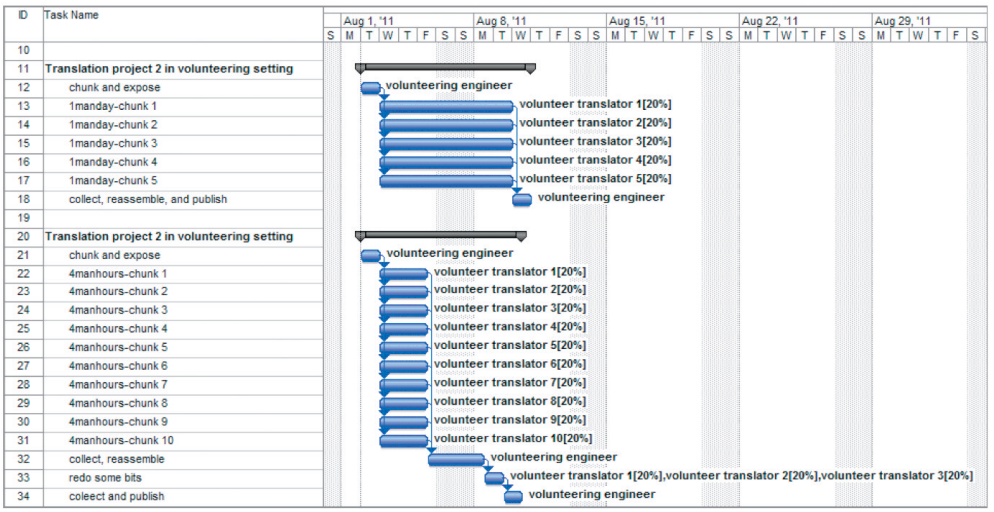

Figure 9: Automated chunking in terms of man-hours is essential for volunteering settings

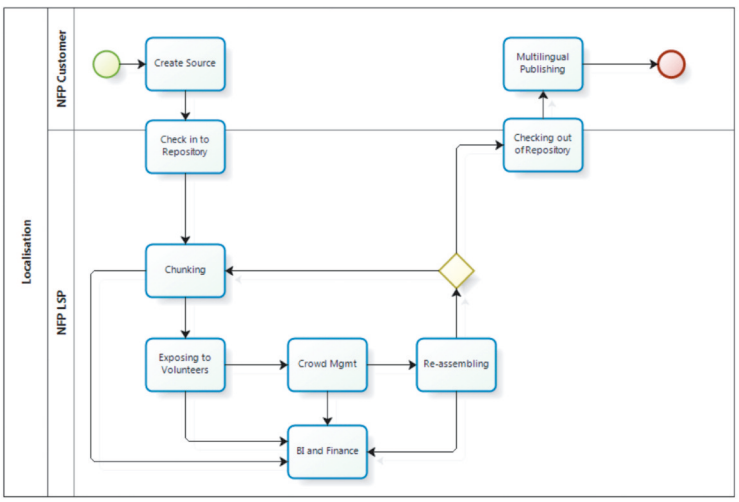

Figure 10: a model of not-for-profit localisation, with further detail provided for content chunking.

The process illustrated in Figure 10 is structurally similar to traditional models. Yet, there are different business needs for the supporting technology between the two different scenarios. There are radical differences, for example, in the availability of resources. In the self-service scenarios that leverage crowd-sourced translation, whether in an enterprise support or a charitable NFP scenario, automated chunking, pull-driven automated assignments, and automated reassembling are a must due to the demand for much finer granularity of chunking. In contrast, in the traditional bulk localisation scenario these are only tentative activities that are often simply performed manually.

6. Conclusion

What is the token and/or the message in the localisation process? We have hinted that ideally the localisation ESB message should have the form of a flexibly chunkable and reassemblable Bitext. With OASIS XLIFF, the industry has such a standard, yet evolving, format to capture industry wisdom and address new business needs. It is capable of carrying payload and metadata with a wide range of granularities and process requirements. Through the business process management practices applied in this paper, we have found that the common denominator of all localisation processes may be as follows:

Parsing of source text -> routing Bitext -> enriching Bitext -> quality assuring Bitext -> exporting target text.

For performing the localisation processes in any organisational setting it is critical to be able to extract global business intelligence from most of the workflows and processes involved.

For an enterprise, managing Bitext has also traditionally meant enforcing process and technology. We argue that this is not a priori a consequence of including Bitext Management in the enterprise process. Rather, in the past, the enterprise may have had to take stringent control due to the lack of standardisation in the areas of both Bitext and Bitext Transformation processes.

Today many enterprise-level practitioners have seen that enforcing process and methodology is not sustainable and/or indeed very expensive. We can see two complementary trends:

1) Standardisation of Bitext message, both payload and metadata.

2) Reuse of available SOA architectures and extra-Localisation workflow solutions, namely the underlying ESBs.

What can be used as the ESB in this case? While most readily-available ESB specialised middleware comes to mind, it can, theoretically, be any sufficiently expres[s]ive3 workflow engine. 'Theoretically' must be emphasised here, as clearly any Turing-complete engine can do what is needed, which is, however, far from claiming that the level of effort needed would be practically achievable or otherwise relevant. In real life situations, many factors play important roles in making this decision, including but not limited to:

1) Legacy investment into and the present state of the overall IS in the organisation

2) Level of fitness of the current IS for the business needs of the organisation

3) Legacy investment into and the present state of specialised localisation technology

4) Importance of unified BI on localisation within the organisation

5) Licensing models of legacy solutions
6) Long term vendor relationships

Enterprise users want to prevent lock-in and manage quality on an 'as needed' basis, which very often applies to string level. In fact, we see, from our case study analysis, the community workflow and the enterprise workflow converging.

The 21st century has seen an onslaught of service-oriented architectures, not only in the IT mainstream but also in the localisation and translation industry. Many an industry player has realised that they no longer wish to be locked in to a particular language technology stack, and some have found their Enterprise Service Buses relevant as potential backbones for what they need to achieve in the area of localisation and translation.

It seems clear that the challenge in the localisation and translation industry is not just of process modelling. It is rather a complex Change Management issue that cannot be properly addressed without applying mature Business Process Management techniques.

Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) in the Localisation Research Centre at the University of Limerick.

All BPMN models used in this paper were created using the free Bizagi Modeler Software, v2.0.0.2, which is BPMN 2 compliant.

References

Aalst, Van der, W.M.P., Hofstede, ter, A.H.M., Kiepuszewski, B., Barros, A.P. (2003) 'Workflow Patterns', Distributed and Parallel Databases, 14, 5-51.

Aalst, Van der, W.M.P. (2004) 'Pi calculus versus Petri nets: Let us eat "humble pie" rather than further inflate the "Pi hype"', unpublished.

Ågerfalk, P. J. and Fitzgerald, B. (2008) 'Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy', MIS Quarterly, 32(2), 385-409.

Bleistein, S., Cox, K., Verner, J. and Phalp, K. (2006) 'Requirements engineering for e-business advantage', Requirements Engineering, 11(1), 4-16.

Blood, R. (2004) 'How Blogging Software Reshapes the Online Community', Communications of the ACM, 47(12), 53-55.

Brabham, D. C. (2008) 'Moving the crowd at iStockphoto: The composition of the crowd and motivations for participation in a crowdsourcing application', First Monday, 13(6).

Business Process Model and Notation v2.0 (BPMN) (2011) Needham, Massachusetts: Object Management Group, Inc (OMG).

Cronin, M. (2010) 'The Translation Crowd', Tradumatica: traduccio i tecnologies de la informacio i la comunicacio, 8, December 2010.

Cox, K., Phalpc, K. T., Bleisteina, S. J. and Vernerb, J. M. (2005) 'Deriving requirements from process models via the problem frames approach', Information and Software Technology, 47(5), 319-337.

DePalma, D.A. (2006) Localization Maturity Model, Lowell: Common Sense Advisory, 16 August.

DePalma, D.A. (2011) Localization Maturity Model 2.0, Lowell: Common Sense Advisory, 28 March.

Galliers, R. D. (2004) 'Reflections on Information Systems Strategizing', in Avgerou, C., Ciborra, C. and Land, F., eds., The Social Study of Information and Communication Technology: Innovation, Actors, and Contexts, Oxford: Oxford University Press, 231-262.

Galliers, R. D. (2006) 'On confronting some of the common myths of Information Systems strategy discourse', in Mansell, R., Quah, D. and Silverstone, R., eds., The (Oxford) Handbook of Information and Communication Technology, Oxford: Oxford University Press.

Hammer, M. and Champy, J. (1993) Reengineering the corporation: a manifesto for business revolution, New York: HarperBusiness.

Howe, J. (2006) 'The rise of crowdsourcing', Wired, 14(6), available: http://www.wired.com/wired/archive/14.06/crowds.html [accessed 4 July 2011].

Jiménez-Crespo, M. A. (2010) 'Web Internationlisation strategies and translation quality: researching the case of "international" Spanish', Localisation Focus, 9(1), 13-25.

Kelly, N., Stewart, R.G. (2011) The Top 50 Language Service Providers, Lowell: Common Sense Advisory, 31 May.

Laudon, K.C. and Laudon, J.P. (1996) Management Information Systems: Organization and Technology, Englewood Cliffs: Prentice-Hall.

Lenker, M., Anastasiou, D. and Buckley, J. (2010) 'Workflow Specification for Enterprise Localisation', Localisation Focus, 9(1), 26-35.

Lewis, D., Curran, S., Doherty, G., Feeney, K., Karamanis, N., Luz, S. and McAuley, J. (2009) 'Supporting Flexibility and Awareness in Localisation Workflow', Localisation Focus, 8(1), 29-38.

Menge, F. (2007) 'Enterprise Service Bus', Free and Open Source Software Conference 2007 (FOSS4G), Victoria, Canada, 24-27 September.

Morera, A., Aouad, L, Collins, J.J. (2011) 'Assessing Support for Community Workflows in Localisation', accepted for 4th Workshop on Business Process Management and Social Software (BPMS2'11), August.

MultiLingual (2011) MultiLingual 2011 Resource Directory, Sandpoint: MultiLingual, available: https://www.multilingual.com/downloads/2011RDPrint.pdf [accessed 1 August 2011].

OASIS XLIFF 1.2 (2008) Oasis, available: http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html [accessed 2 August 2011].

O'Hagan, M. (2009) 'Evolution of User-generated Translation: Fansubs, Translation Hacking and Crowdsourcing', The Journal of Internationalisation and Localisation, 1, 94-121.

Paulk, M.C., Curtis, W., Chrissis, M.B., Weber, C.V. (1993) Capability Maturity Model for Software, Version 1.1, Technical Report CMU/SEI-93-TR-024 ESC-TR-93-177, Pittsburgh, Pennsylvania: Software Engineering Institute, available: http://www.sei.cmu.edu/reports/93tr024.pdf [accessed 2 August 2011].

Sargent, B.B., DePalma, D.A. (2008) Translation Management Systems, Lowell: Common Sense Advisory, 16 September.

Sargent, B.B., DePalma, D.A. (2007) Translation Management System Scorecards, Lowell: Common Sense Advisory, 26 February.

Turban, E., Leidner, D., McLean, E. and Wetherbe, J. (2007) Information Technology for Management: Transforming Organizations in the Digital Economy, ed. 6, Wiley.

WordPress.org (2011a) Showcase [online], available: https://wordpress.org/showcase/ [accessed 13 July 2011].

WordPress.org (2011b) Roles and Capabilities [online], available: https://codex.wordpress.org/Roles_and_Capabilities [accessed 5 July 2011].

Yunker, J. (2003) Beyond Borders: Web Globalization Strategies. Indianapolis: New Riders.

Enabling Complex Asian Scripts on Mobile Devices

Waqar Ahmad
Computer Science Department,
National University of Computer and Emerging Sciences, Lahore, Pakistan
waqar.ahmad@nu.edu.pk

Sarmad Hussain
Center for Language Engineering,
Al-Khawarzimi Institute of Computer Science,
University of Engineering and Technology,
Lahore, Pakistan
sarmad@cantab.net

Abstract

The increasing penetration of mobile devices has resulted in their use in diverse domains such as education, health, entertainment, business, sports, and social networks. However, a lack of appropriate support for many local languages on mobile devices, which use complex scripts rather than Latin scripts, is constraining many people across developing Asia and elsewhere from using their mobile devices in the same way. There are some ad hoc solutions for certain scripts, but what is needed is a comprehensive and scalable framework which would support all scripts. The Open Type Font (OTF) framework is now being widely used for supporting complex writing systems on computing platforms. If support for OTF is also enabled on mobile devices, it would allow them to also support complex scripts. This paper reports on work in this area, taking Pango, an open source rendering engine, and porting and testing its language modules on a mobile platform to provide support for Open Type Fonts. The paper describes the process for successful deployment of this engine on Nokia devices running the Symbian operating system for Urdu, Hindi and Khmer languages. The testing results show that this is a viable solution for enabling complex scripts on mobile devices, which can have significant socio-economic impact, especially for developing countries.

Keywords: *: Mobile Devices, Smart-Phones, Pango, Localisation, Open Type Fonts, Complex Writing Systems*

1. Introduction

The number of mobile phone subscriptions worldwide is expected to reach 5 billion in 2010 (ITU 2010). Mobile device penetration in developing countries of Asia is also increasing at a rapid pace (MobiThinking 2010). While current and past usage of mobile devices has mostly been for voice, there is a significant increase in text and other data services using smart-phones (adMob 2010). It is expected that more than 85% of mobile handsets will be equipped for mobile web access by the end of 2011 (MobiThinking 2010), as many smart-phones today have processing power and other capabilities comparable to desktop computers of the early 1990s.

As the hardware capabilities of mobile devices improve, they are increasingly being used in areas like education, health, entertainment, news, sports, and social networks. This usage of smart-phones requires that text and other data services are made available in local languages. However, most of the mobile devices that are currently in use only support Latin script. There is limited or no support available for many other language scripts, specifically those of developing Asia. The devices generally support basic Latin, bitmap and True Type Fonts (TTF). Most Asian languages scripts, on the other hand, are very cursive, context sensitive and complex (Hussain 2003; Wali and Hussain 2006), and can only be realized using more elaborate font frameworks, e.g. Open Type Fonts (OTF) (Microsoft 2009). Such frameworks are not supported on most mobile devices and smart-phones at this time. Many people in developing Asia are only literate in their own languages and are, therefore, unable to utilize their mobile devices for anything other than voice calls. Developing font support is an essential pre-cursor to making content available in local language scripts. Once support is in place, content can be created, allowing people to utilize the additional capabilities of mobile phones for their socio-economic gains.

Whether focusing on iPhone (Apple Inc. 2010), Symbian based Nokia Phones (Forum.Nokia Users 2009), Google Android (Google 2009), Windows Mobile (Microsoft 2010), or Blackberry, the worldwide web is full of queries and posts showcasing the needs and concerns of developers and end-users, who are looking for particular language support on their devices. While there is extensive localisation support for desktop computers, mobile

devices are lagging behind. Smart-phone software developers try to find workarounds for resolving localisation issues and sometimes achieve limited success. However, total success can only be achieved if the underlying device platform provides comprehensive support. If the underlying platform has limitations, then they are also reflected in the workarounds produced by software developers. A major problem is that mobile platforms provide limited software internationalisation support and therefore, localisation for certain languages may become very difficult.

In this paper we have suggested a solution for alleviating some of the problems associated with the support of complex Asian scripts on mobile devices using Pango - an open source library for text layout and rendering with an emphasis on internationalisation (Taylor 2004). Research and development has been carried out with a focus on evaluating the viability of Pango as a text layout and rendering engine on mobile platforms. For this project, Symbian has been chosen as the mobile platform. The project has two components: one component deals with porting script specific modules of Pango to the Symbian platform; the other component is the development of an application (referred to as the SMSLocalized Application hereinafter) that can send/receive SMS in local languages using Pango with mobiles, as a proof of concept.

Although all of the language specific modules of Pango have been successfully ported to the Symbian platform, extensive testing is performed for Urdu and an initial level of testing is performed for Khmer and Hindi. The results of the tests are quite promising and confirm the viability of Pango as a font engine for mobile devices. The SMSLocalized application contains features customised for local language scripts. This application has been tested for Urdu; however, the architecture of the application is very flexible and as such allows quick application customization for other languages. This paper presents the relevant background and details of this work.

2. Current Localisation Support on Mobile Platforms

Limitations in script support on mobile devices are often due to constraints specific to mobile handsets such as a small amount of memory, limited processing power and other factors. During our research, we have learnt that most of the issues related to localisation on mobile phones fall into one or more of following patterns:

- The localisation features supported on a mobile device may not be adequately documented. As a result of this, information about localisation features may only become known after acquiring and evaluating the device by installing localised software.

- Only a limited set of features for a language may be supported on the device. For instance, True Type Fonts (TTF) may be supported but not Open Type Fonts (OTF), which will results in lack of support of a various languages and their scripts.

- In mobile device system software, language support may exist at the level of menu items but may be missing at application level. For instance, a device may have an operating system with a properly localised user interface but an on-device messenger application may not allow the user to input text in a local language.

- A particular device platform may support many languages as a whole. However, when a device is released into the market, it may only be equipped with a subset of the platform's supported languages. For instance, a language-pack may be missing or the font rendering engine may be constrained by its multilingual language support.

As previously mentioned, software developers continue trying to find workarounds for the localisation issues which are, in many ways, limited by the support provided by the underlying device platform. The following sections give an overview of the extent of localisation support on some of the major smart-phone platforms.

A. Symbian

Symbian OS, currently owned by Nokia, is the most widely deployed operating system on mobile phones. It supports application development using Java Micro Edition (Java ME) and C/C++. Symbian operating system supports a very basic level of user interface which does not make it usable by layman users. Therefore, on top of the Symbian operating system, some mobile device vendors have developed rich user interfaces. Two such user interfaces are S60, developed by Nokia, and UIQ, developed by UIQ technology. (Morris 2007).

Symbian supports a number of languages. However, it does not support Open Type Fonts (Forum.Nokia 2009). Its default engine is based on the FreeType font library (Forum.Nokia 2009). The Symbian operating system, however, can be extended by plugging in an external font engine to add support for languages or scripts not already supported (Morris 2007). For instance, an engine can be developed, or adapted from open source, that adds support for open type fonts with complex scripts i.e. if a third party developer wants open type font support, s/he can develop and plug the font engine into the operating system which can then be used by any software application on the device.

B. Windows Mobile

Windows Mobile is a Windows CE based operating system developed by Microsoft. Windows CE is primarily designed for constrained devices like PDAs and can be customized to match the hardware components of the underlying device (Microsoft 2010). Windows Mobile supports the Microsoft .Net Compact Framework for application development, which in turn supports a subset of Microsoft .Net Framework features.

According to the Microsoft website (Microsoft 2010), WordPad, Inbox, Windows Messenger, and File Viewer applications are not enabled for complex scripts like Arabic, Thai, and Hindi.

There are some commercial solutions for localisation on the Windows Mobile platform. One such solution is Language Extender. It supports Arabic, Czech, English, Estonian, Farsi, Greek, Hebrew, Hungarian, Latvian, Lithuanian, Polish, Romanian, Russian, Slovak, and Turkish (ParaGon Software Group 2010). However, Open Type Fonts for other complex writing systems, e.g. Urdu Nataleeq (Wali and Hussain 2006) are not available.

C. Android

Android is a relatively new mobile software stack based on Linux. It allows application development using the Java programming language. However, a native SDK is also available from the Android developer website that can be used to develop native applications in C/C++ (Google 2010).

Localisation on the Android platform is still limited to a few languages. Independent developers have tried workarounds with limited success (Kblog 2009). There is lot of debate on language support issues on Android forums (Google Android Community 2010). However, it has still not been made clear, officially, from Google as to when support for OTF will be included.

Google (2009) talks about localisation for German, French, and English but does comment about languages using non-Latin scripts.

D. Apple iPhone

According to Apple (Apple 2010), the Apple iPhone 3G supports a number of languages including English (U.S), English (UK), French (France), German, Traditional Chinese, Simplified Chinese, Dutch, Turkish, Ukrainian, Arabic, Thai, Czech, Greek, Hebrew, Indonesian, Malay, Romanian, Slovak, and Croatian. Again, only TTF based fonts, e.g. for Arabic script, are supported, and OTF fonts are not supported.

E. Monotype Imaging Rasterization and Layout Engines for Mobile Phones

Monotype imaging (2010) provides engines for font rasterization (iType Font Engine) and layout (WorldType Layout Engine) for smart-phones. The solution is ANSI C based and is available for integration with Android, Symbian and Windows CE. However, full Open Type Font support is not available in their solutions.

F. Other Smart-phone Platforms

Other smart-phone platforms like RIM Blackberry, Palm WebOS etc. are not investigated in detail from a localisation perspective in the current work. They support localisation features, however, their limitations are similar to those mentioned above, as are discussed on online developer and end-user forums (ParaGon Software Group 2010).

3. Current Work

An investigation is conducted to evaluate the possibility of using Pango as a text rendering and layout engine for smart-phones. The project covers the following:

1. Porting language specific modules of Pango to the Symbian operating System.

2. Development of an SMS application (SMSLocalized), designed so that it can be customized for scripts supported by Pango.

As Symbian is a dominant and mature mobile platform, it has been chosen for this project. Pango

has a basic module and multiple scripts for specific modules, e.g. for Arabic/Urdu, Indic, Khmer, Tibetan, etc. There has already been a compilation of Pango for the Symbian platform (Cairo Graphics 2009), however, this compilation only covers the basic module, and script-specific modules have not been ported. We use Cairo and compile individual script modules on Symbian. Among the modules ported, Arabic (for Urdu), Indic and Khmer are tested after deployment. The rest of the paper is focused on this process of porting and testing the script specific modules of Pango on the Symbian platform.

A. Symbian Overview

As said earlier, Symbian OS is currently the most widely deployed operating system on mobile phones. It supports application development using Java and C/C++. Java application development on Symbian is enabled using Java Micro Edition (Java ME) and C/C++ application development is enabled using the native OS application framework. (Morris 2007). To fully exploit native device features, development in C/C++ is required. Therefore, for this project, which requires extensive native device features, the development is also carried out in C/C++. A typical Symbian C/C++ application is designed according to Model-View-Controller (MVC) architecture (Harrison and Shackman 2007). The SMSLocalized Application has also been developed according to the same MVC architecture.

As Pango is a C based library (Martensen 2009), Symbian support for C/C++ makes it easier to port the library. Depending upon the type of features accessed by an application from the device operating system, a Symbian application may require official signing from Symbian Signed. For development and testing of our application, we used the 'developer certificates.'

B. Pango Overview

Pango is a popular text layout and rendering library used extensively on various desktop platforms. Pango is the core library used in GTK+-2.x for text and font handling (Martensen 2009; also Taylor 2004). Pango has a number of script specific modules, including modules for Arabic, Hebrew, Hangul, Thai, Khmer, Syriac, Tibetan, and Indic scripts. Pango can work with multiple font back-ends and rendering libraries as mentioned in the following list (Martensen 2009).

- Client side fonts using the FreeType and Fontconfig libraries. Rendering can be done with Cairo or Xft libraries, or directly to an in-memory buffer with no additional libraries.

- Native fonts on Microsoft Windows using Uniscribe for complex-text handling. Rendering can be done via Cairo or directly using the native Win32 API.

- Native fonts on MacOS X using ATSUI for complex-text handling. Rendering using Cairo. ATSUI is the library for rendering Unicode text on Apple Mac OS X.

C. R&D Challenges

Mobile application development poses a lot of challenges primarily due to the constrained nature of the devices. Limited memory size, low processing power, dependency on batteries, constrained input and output modalities and limited system API access, are just some of the many constraints faced by mobile application developers and researchers.

While the support for high level application development for mobile devices is extensively available, low-level application development remains challenging. Even more challenging is exploring areas which are relatively lesser traversed by application developers and researchers e.g. localisation and font rendering. Lack of documentation, few forum discussion threads, scarcity of expert developers, the unpredictable nature of development and the limited debugging and testing platforms, are among some of the major challenges that we faced during project R&D on localisation for smart-phones. Even installation of a font file on a mobile device may at times become a challenge. For example, it is not always easy to find out where to copy font files, how to get the device to detect a new font etc. Details such as these may only be known after extensive exploration of the device platform under consideration, as it may be documented well for application developers.

D. Libraries

The integration of Pango with Cairo provides a complete solution for text handling and graphics rendering. The combination of Pango and Cairo, along with their dependencies, is compiled for the Symbian platform as part of this project. The following libraries are required for complete solution to work properly:

1) Pango

Pango is a font rendering and text layout engine

available with an open source license. Pango has a number of language specific modules, including modules for Hebrew, Arabic, Hangul, Thai, Khmer, Syriac, Tibetan, and Indic scripts (Martensen 2009), as discussed.

2) Cairo

Cairo is a 2-D graphics library which supports multiple output devices i.e. X-Window, Win32, PDF, SVG etc. The library has been written in the C programming language; however, its bindings are available in other languages such as Java, C++, PHP etc. (Cairo Graphics 2010).

3) FreeType

FreeType is an ANSI C compliant font rasterization library. It provides access to font files of various formats and performs actual font rasterization. Font rasterization features include the conversion of glyph outline of characters to bitmaps. It does not provide APIs to perform features like text layout or graphics processing (Free Type 2009).

4) FontConfig

FontConfig allows the selection of an appropriate font given certain font characteristics. It supports font configuration and font matching features and depends on the Expat XML parser. FontConfig has two key modules: The Configuration Module builds an internal configuration from XML files and the Matching Module accepts font patterns and returns the nearest matching font (FontConfig 2009).

5) GLib

GLib provides the core application building blocks for libraries and applications written in C. It provides the core object system used in GNOME, the main loop implementation, and a number of utility functions for strings and common data structures (Pango 2009).

6) Pixman

Pixman is a low level pixel manipulation library for X and Cairo. Supported pixel manipulation features include image compositing and trapezoid (Pixman 2009).

7) Expat

Expat is an XML parsing library written in C. It is a stream-oriented parser in which an application registers handlers for components that the Expat parser might find in the XML document e.g. XML start tags (Expat 2009).

8) libpng

Libpng is a library written in C for the manipulation of images in PNG (Portable Network Graphics) format (Roelof 2009).

E. Tools and Technologies

The following tools and technologies are used for the development of this work.

1) Code Baseline

Code from http://code.google.com/p/cairo-for-symbian/ (Cairo Graphics 2009) is taken as baseline for the current work. This is an earlier compilation of the basic Pango module for the Symbian platform.

2) Development Tools

The Following tools were used during development:

- Carbide C++ v2.3.0: an IDE provided by Nokia for application development on the Symbian platform (Forum.Nokia 2009).
- Symbian S60 3rd Edition Feature Pack 2 SDK v1.1.2: a development kit for Nokia S60 and Symbian platforms. It includes a simulator for testing applications on a Windows desktop before they are installed and tested on actual devices (Forum.Nokia 2009).

F. Application Architecture

The project has two major parts. The first is an SMS application for testing font support and porting of the language modules of Pango and development.

1) SMSLocalized Application

The SMSLocalized application is a Symbian application designed for the languages supported through Pango. The application has the following features.

- Allows typing of text using an SMS Text editor.
- Displays an on-screen keypad, which is configurable based on a text-file for a language.
- Sends and receives text as SMS.
- Automatically wakes up whenever a new message is received.

The SMSLocalized application is implemented for the Urdu language, chosen for its complexity in contextual shaping and positioning of glyphs (Hussain 2003).

Figure 1 depicts the SMSLocalized application class diagram developed in Symbian C/C++. SMSLocalized Application, SMSLocalizedDocument, SMSLocalizedAppUi, and NewMessageContainerView are required by the MVC architecture of Symbian applications.

To enable Urdu text input on mobile phones, a custom key map has to be defined so that the appropriate Urdu characters are rendered against each key press. Many mobile phones support multi-tapped text input, where each key on the keypad represents more than one characters. This arrangement of character sequences against each numeric key on the mobile phone is called the keymap i.e. each numeric key on the device has an associated keymap.

On a typical Symbian device, a keymap is defined against each key on the device keypad so a character can be entered using the multi-tapping nature of Numeric keypads. NumerciLocalizedPtiEngine provides customized low level input mechanisms. One key feature supported in this class is that it defines a new keymap for the local language. NumericKeypad is used to draw a custom localised keypad on the mobile screen. This involves measuring screen size and dividing it appropriately to allow sufficient space for a numeric keypad consisting of four rows and three columns while still giving enough space to enter text. The CSMSWatcher class inherits from CActive and registers an active object with the scheduler. It implements methods to handle messages received by the application.

To prevent the Symbian operating system from loading the default keymap and using the customized keymap for another local language, a new keymap has to be defined and a mechanism developed to load this sequence of characters when the application starts up. This involves defining a custom Unicode sequence against each key on the numeric keypad in a text file and using the CPtiEngine API of the Symbian platform to load customized keymap sequences from the relevant resource file.

2) Script Specific Modules of Pango

The second major component of the solution is the Pangocairo library core and script-specific modules. The Pangocairo library, along with script-specific modules, are compiled and ported to Symbian platform.

Pango supports multiple scripts including Latin, Cyrillic, Arabic, Hangul, Hebrew, Indic and Thai. Figure 2 provides an overview of the high level architecture of Pango (Taylor 2001). The following are key features of the Pango Architecture (Taylor 2001):

- Unicode has been used as common character encoding mechanism throughout the Pango system.
- There is a core functionality module, Pango Core, which includes functions such as itemization (subdivision of text strings) and line breaking.
- There are script specific modules for handling features unique to each script. Each script module

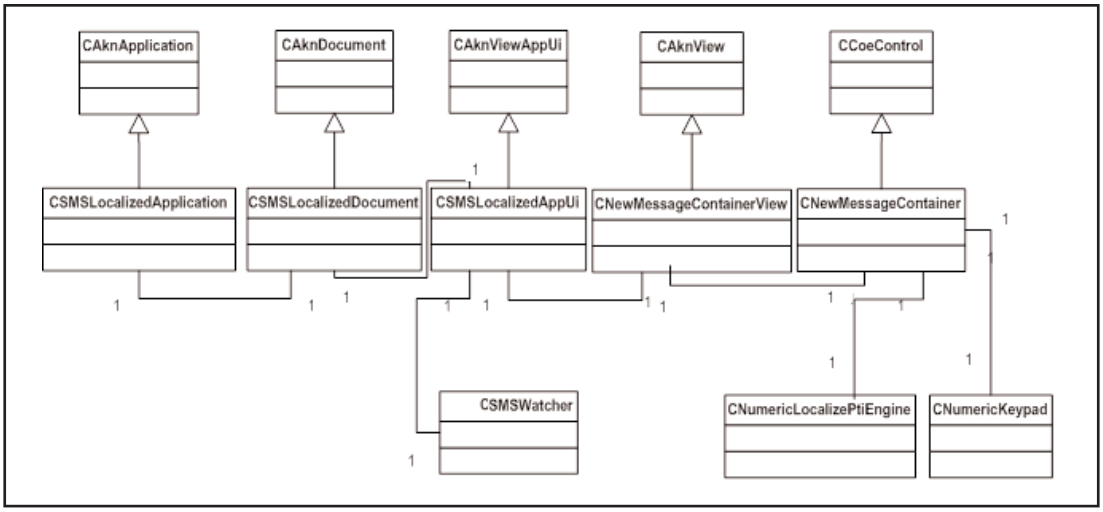

Figure 1: Class Diagram of the SMSLocalized Application

has been further split into two modules: the language module and the shaper module. The language module is independent of the rendering system and the shaper module (e.g. Arabic X Shaper, PS X Shaper) is dependent on the rendering system.

- Pango rendering components support multiple rendering back ends. There are separate components for each rendering backend e.g. X rendering backend is responsible for rendering X fonts using XLib and XServer.

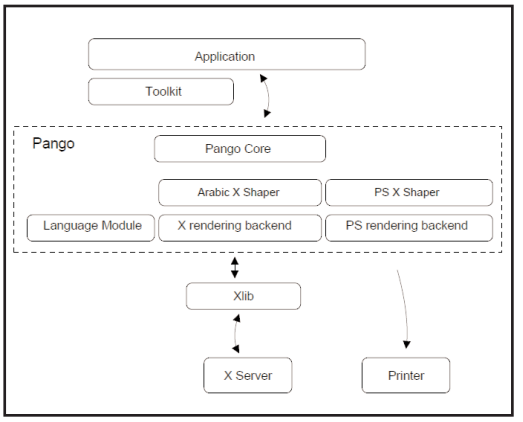

Figure 2: Pango Architecture (Taylor 2001)

Pangocairo itself includes packages of standard C/C++ libraries. Therefore, it can be ported to the Symbian platform, which also supports C/C++. However, this task is challenging because the availability of the technical information required is limited. The following are some important modifications carried out in Pango and its dependent libraries in order to port it onto the Symbian operating system.

- Declarations of language specific modules are included in the code, which lead to the generation of interface functions. These interface functions enable access to the language specific modules in the code.

- The source code that needs to be compiled for the Symbian operating system must be referred to in appropriate 'project make files' i.e. .mmp files. References to interface components of script specific modules (e.g. Arabic) are included in appropriate .mmp files.

- On start-up, the Symbian operating system loads font files from specific folders. Since the FontConfig library accesses font files, it is updated so that it can access Nafees Nastalique font files loaded by the Symbian operating system.

- Some of required Pango API functions are not exposed for external access in the Symbian code. Such functions are declared and listed in appropriate interface files.

In addition to the above, a component that interfaces with the Pango library has been created. This component enables access to the text rendering features of Pango i.e., it can take any Unicode text as input and return the rendered text in a format compatible with the requirements of the Symbian operating system.

3) Deployment and Testing Platforms

Both components of the solution were deployed and tested on the following platforms.

- WINSCW

This is a simulator for theS60 Symbian platform included in Symbian S60 3rd Edition Feature Pack 2 SDK v1.1.2 for Windows Platform.

- Nokia E51 (A Symbian Phone)

The following are the specifications of the Nokia E51 handset-a Symbian based phone:

i. Symbian: v9.2 S60 v3.1 UI
ii. CPU: ARM 11 369 MHz Processor
iii. RAM: 96 MB
G. Testing Results

The SMSLocalized application and language specific modules of Pangocairo framework were deployed and tested on both a Windows emulator (Symbian S60 3rd Edition) and a real device (Nokia E51). The application works successfully on both platforms. Figure 3 shows the SMSLocalized application running on the Nokia S60 3rd Edition Emulator. The on-Screen Urdu Keypad in Nafees Nastalique Open Type Font can also be seen. Figure 4 shows Urdu text written in Nafees Nastalique font (an Open Type Font) as rendered on the Nokia E51.

An Open Type Font file contains glyphs and rules. The glyph tables are in a similar format to those used to store vectorized outlines for TTF files. In addition, rules for glyph positioning and their contextual substitution are represented in different tables.

Finally, marks which are associated with glyphs can also be adjusted through rules for finer tuning of fonts. All of these aspects are thoroughly tested for Nafees Nastalique, and the open Urdu font freely available online. More than 500 Urdu ligatures[1] consisting of two to eight characters are chosen from the list of valid ligatures available online (CRULP 2009). The arbitrary selection includes complex ligatures, which exhibit cursiveness, context sensitive shaping and positioning of glyphs. Table 1 shows the ligature counts for two to eight character combinations selected for this testing.

The ligature set included all available Urdu characters.

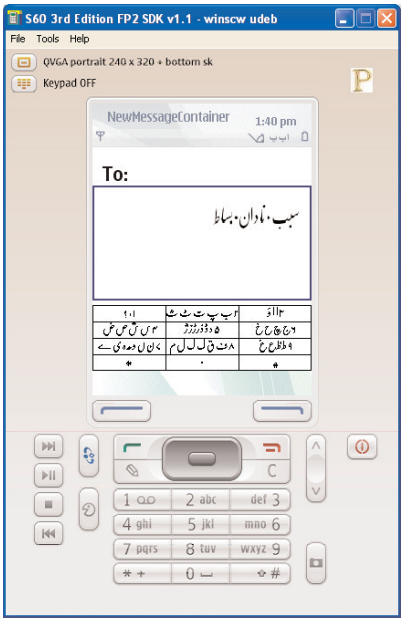

Figure 3: the SMSLocalized Application on Nokia S60 3rd Edition Emulator.

Character Count per Ligature	Number of Ligatures Tested
2	90
3	107
4	95
5	81
6	98
7	65
8	20

Table 1: Summary of Ligature Set Selected for Testing

Table 2 shows the frequency of each letter in the test set and the contexts (initial, medial, final and isolated) in which it has been tested. In addition, the mark association and placement is tested. Though the current tests do not test every possible shape of each Urdu letter, as there is glyph variation based on other characters in the context and not just the four contexts listed, the testing is still representative and these results can be extrapolated to un-tested substitution and positioning rules with confidence. The shaded cells in the table are for non-joining characters, which do not occur in initial or medial positions. The ligatures were displayed and manually tested on the Symbian S60 Emulator (WINSCW) and the Nokia E51device.

Figure 4: Pango Urdu (Open Type Font Nafees Nastalique) text rendering on a Nokia E51

Figures 4 and 5 show the rendering results of some of the selected ligatures on the phone and emulator respectively, showing the cursiveness, glyph substitution, glyph positioning and mark placement complexities.

[1] Ligature is the portion of the written representation of a word that is formed by characters combining together. A word may have one or more ligatures and a ligature may be formed by one or more characters. A non-joining character or a word-ending will end a ligature.

Character	Frequency	Context			
		Initial	Medial	Final	Isolated
ا	131			97	20
ب	115	65	36	2	12
پ	113	39	63	4	7
ت	177	13	135	18	11
ٹ	102	18	71	4	9
ث	11	2	3	3	3
ج	53	33	17	1	2
چ	69	30	33	3	3
ح	26	11	9	5	1
خ	13	5	4	2	2
د	18			11	7
ڈ	14			10	4
ذ	7			6	1
ر	18			13	5
ڑ	4			2	2
ز	7			3	4
ژ	4			3	1
س	90	24	58	3	5
ش	35	16	8	7	4
ص	14	4	7	1	2
ض	9	3	2	2	2
ط	19	6	9	2	2
ظ	11	3	5	2	1
ع	22	5	12	2	3
غ	12	5	4	2	1
ف	26	9	13	1	3
ق	14	4	6	1	3
ک	98	24	61	2	11
ک	61	23	30	3	5
ل	138	26	101	5	6
م	80	31	35	5	9
ن	254	42	172	19	21
ں	9			6	3
و	21			13	8
ہ	69	7	17	26	19
ھ	176	3	168	5	
ء	5				5
ی	308	6	228	61	13
ے	131			119	12

Table 2: Context and Distribution of Urdu Characters in the Test Set of 500 Ligatures

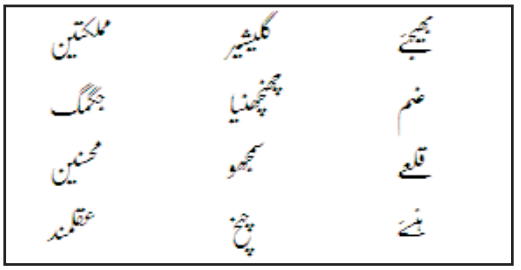

Figure 5: Pango Urdu (Open Type Font Nafees Nastalique) text rendering on Nokia S60 Emulator

After display, all the ligatures were manually inspected for correct shaping, substitution and mark placement. Where there are potential ambiguities, the same are compared with the rendering on the computer to see whether it is the source rendering or the font rules. Detailed testing shows that there are no errors which can be attributed to the porting of these script-specific modules of Pango, verifying completely accurate porting for the module for Arabic script as used for the Urdu language.

The Khmer and Indic modules have also been compiled and tested using limited text. Though no errors have been found, more extensive testing is required for complete verification, so these testing details are not reported at this time. Figure 6 shows Urdu, Devanagari (using the Indic module), and Khmer rendered on Symbian S60 3rd edition emulator.

Figure 6: Urdu, Devanagari, and Khmer rendered on Symbian S60 3rd edition emulator.

4. Conclusion

The global penetration of smart-phones is making local language support for them both urgent and significant, as an increasing number of mobile users want the devices to access local language content. However, we have learnt that smart-phones are still far from current desktops in their support for the local language scripts of developing Asia. The Symbian platform, among the oldest and mature mobile platforms, does not provide complete Open Type

Font (OTF) support. However, the porting of Pango script-specific modules can add OTF support to Symbian. This has been successfully achieved through our project. All of the Pango language script modules have been ported to the Symbian OS, with extensive testing carried out for Urdu and initial testing performed for Khmer. Through the process, we have learnt that the Urdu, Indic and Khmer language modules of Pango work well on the Symbian platform. We believe that given the extensive support for international languages by Pango, it is a good choice for serving as a text layout and rendering engine for smart-phone devices.

Currently, the project is continuing to port and test additional script modules. The SMSLocalized application is being integrated to communicate with Pango for rendering and additional work is underway to develop similar support for the Android open source platform.

Acknowledgements

This work has been supported by the PAN Localization project (www.PANL10n.net) grant by IDRC Canada (www.idrc.ca), administered through Center for Language Engineering (www.CLE.org.pk), Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan.

References

adMob (2010) AdMob Mobile Metrics [online], available: http://metrics.admob.com/ [accessed 15 Aug 2010]

Android Developers on Google Groups (2010) Localization [online], available: http://groups.google.com/group/adroid-platform/browse_thread/thread/8887a2fe29c38e7 [accessed 17 Aug 2010].

Apple (2010) iPhone 4 Technical Specifications [online], available:http://www.apple.com/iphone/specs.html [accessed 20 Aug 2010].

Cairo Graphics (2010) Cairo Tutorial [online], available: http://cairographics.org/tutorial/ [accessed 12 May 2009].

Cairo Graphics (2009) Cairo for Symbian OS [online], available: http://code.google.com/p/cairo-for-symbian/ [accessed 18 May 2009].

CRULP (2009) Valid Ligatures for Urdu [online], available: http://www.crulp.org/software/ling_resources/UrduLigatures.htmhttp://www.forum.nokia.com/ [accessed 11 Mar 2010].

Edwards, L. and Barker, R. (2004) Developing S60 Applications: A Guide for Symbian OS C++ Developers, U.S.: Addison Wesley.

Expat (2009) The Expat XML Parser [online], available: http://expat.sourceforge.net/ [accessed 13 May 2009].

Free Type (2009) The FreeType Project [online], available: http://www.freetype.org/index2.html [accessed 12 May 2009].

FontConfig (2009) User's Manual [online], available: http://fontconfig.org/fontconfig-user.html [accessed 13 May 2009].

Forum.Nokia (2009) Support for Open Type Fonts [online], available: http://discussion.forum.nokia.com/forum/showthread.php?163031-Support-for-Open-Type-Fonts [accessed 16 Aug 2010].

Forum.Nokia Users (2009), Discussion Board [online], available:http://discussion.forum.nokia.com/forum/ [accessed 7 Oct 2009].

Google (2009) Localizing Android Apps [DRAFT] [online], available:http://groups.google.com/group/android-developers/web/localizing-android-apps-draft [accessed 14 May 2010].

Google (2010) Android 2.2 Platform [online], available: http://developer.android.com/sdk/android-2.2.html [accessed 10 Oct 2010].

Google Android Community (2010) Arabic Language Support [online], available: http://code.google.com/p/android/issues/detail?id=5597&colspec=id%20type%20status%20owner%20summary%20stars [accessed 19 Aug 2010].

Harrison, R. and Shackman, M. (2007) Symbian OS C++ for Mobile Phones: Application Development for Symbian OS v9, England: John Wiley & Sons, Ltd.

Hussain, S.(2003). 'www.LICT4D.asia/Fonts/Nafees_Nastalique.' Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society, Asian Media Information Center, Singapore.

International Telecommunication Union (2010) ITU sees 5 billion mobile subscriptions globally in 2010 [online], available: http://www.itu.int/newsroom/press_releases/2010/06.html [accessed 18 Aug 2010].

Kblog (2009) Arabic Language in Android [online] available: http://blog.amr-gawish.com/39/arabic-language-in-android/ [accessed 19 Aug 2010]

Microsoft (2009) OpenType Specification [online], available: http://www.microsoft.com/typography/otspec/ [accessed 10 Oct 2010].

Microsoft (2010) Creating a Complex Scripts-enabled Run-Time Image [online], available: http://msdn.microsoft.com /en-us/library/ee491707.aspx [accessed 16 Aug 2010].

MobiThinking (2010) Global mobile stats: all latest quality research on mobile Web and marketing [online], available: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats [accessed 16 Aug 2010].

Monotype Imaging (2010) Products and Services [online], available: http://www.monotypeimaging.com/products-services/ [accessed 5 Aug 2010].

Morris, B. (2007) The Symbian OS Architecture Sourcebook: Design and Evolution of a Mobile Phone OS, England: John Wiley & Sons, Ltd.

Pango (2009) Pango Reference Library [online] available: http://library.gnome.org/devel/pango/stable/ [accessed 15 May 2009].

ParaGon Software Group (2010) Language Extender for Windows Mobile Pocket PC [online], available: http://pocket-pc.penreader.com/ [accessed 16 Aug 2010].

ParaGon Software Group (2010) PILOC for Palm [online] available: http://palm.penreader.com/ [accessed 24 Aug 2010].

Pixman (2009) Pixmann [online], available: http://cgit.freedesktop.org/pixman [accessed 13 May 2009].

Roelof, G. (2009) LibPng for Windows [online], available: http://gnuwin32.sourceforge.net/packages/libpng.htm [accessed 15 May 2009].

Roelof, G. (2009) LibPng [online], available: http://www.libpng.org/pub/png/libpng.html [accessed 15 May 2009].

Sales, J. (2005) Symbian OS Internals: Real-time Kernel Programming, England: John Wiley & Sons, Ltd.

Taylor, O. (2004) 'Pango, an open-source Unicode text layout engine,' 25th Internationalization and Unicode Confernece, Unicode Consortium, Washington DC.

Taylor, O. (2001) Pango: Internationalized Text Handling [online],available:http://fishsoup.net/bib/PangoOls 2001.pdf [accessed 10 Jun 2009].

Wali, A., Hussain, S. (2006) 'Context Sensitive Shape-Substitution in Nastaliq Writing system: Analysis and Formulation,' Proceedings of International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE2006).

LocConnect: Orchestrating Interoperability in a Service-oriented Localisation Architecture

Asanka Wasala, Ian O'Keeffe and Reinhard Schäler
Centre for Next Generation Localisation
Localisation Research Centre
CSIS Dept., University of Limerick,
Limerick, Ireland.
{Asanka.Wasala, Ian.OKeeffe, Reinhard.Schaler}@ul.ie

Abstract

Interoperability is the key to the seamless integration of different entities. However, while it is one of the most challenging problems in localisation, interoperability has not been discussed widely in the relevant literature. This paper describes the design and implementation of a novel environment for the inter-connectivity of distributed localisation components, both open source and proprietary. The proposed solution promotes interoperability through the adoption of a Service Oriented Architecture (SOA) framework based on established localisation standards. We describe a generic use scenario and the architecture of the environment that allows us to study interoperability issues in localisation processes. This environment was successfully demonstrated at the CNGL Public Showcase in Microsoft, Ireland, November 2010.

Keywords: *: Interoperability, Localisation, SOA, XLIFF, Open Standards*

1. Introduction

The term localisation has been defined as the "linguistic and cultural adaptation of digital content to the requirements and locale of a foreign market, and the provision of services and technologies for the management of multilingualism across the digital global information flow" (Schäler 2009). As the definition suggests, localisation is a complex process. Localisation involves many steps: project management, translation, review, quality assurance etc. It also requires a considerable effort as it involves many languages, dealing with characteristics and challenges unique to each of these languages such as the handling of right-to-left scripts, collation, and locale specific issues. Time-frame is another parameter that affects the complexity of the localisation process. Localisation processes require dealing with frequent software updates, short software development life cycles and the simultaneous shipment of source and target language versions (simship). A broad spectrum of software is required to handle the process, ranging from project management software to translation software. A large number of file formats are encountered during the localisation process. These file formats may consist of both open standard and proprietary file formats. Localisation processes involve different types of organisations (e.g. translation and localisation service providers) and different professions (e.g. translators, reviewers, and linguists). Localisation constantly has to deal with new challenges such as those arising in the context of mobile device content or integration with content management systems. In this extremely complex process, the ultimate goal is to maximise quality (translations, user interfaces etc.) and quantity (number of locales, simships etc.) while minimising time and overall cost.

Interoperability is the key to the seamless integration of different technologies and components across the localisation process. The term interoperability has been defined in a number of different ways in the literature. For example, Lewis et al. (2008) define interoperability as: "The ability of a collection of communicating entities to (a) share specified information and (b) operate on that information according to an agreed operational semantics".

The most frequently used definition for the term "interoperability" is by the IEEE: "Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged." (IEEE, 1991).

However, interoperability, while presenting one of the most challenging problems in localisation, has not had much attention paid to it in the literature. We

aim to address this deficit by presenting a novel approach to interoperability across localisation tools through the adoption of a Service Oriented Architecture (SOA) framework based on established localisation standards. We describe a generic use scenario and the architecture of the approach offering an environment for the study of interoperability issues in localisation process management. To our knowledge, this is the first demonstrator prototype based on SOA and open localisation standards developed as a test bed in order to explore interoperability issues in localisation.

The remainder of the paper is organized as follows: Section 2 provides an overview of interoperability in general, and in localisation in particular, in the context of open localisation standards; Section 3 explains the experimental setup, introduces the LocConnect framework, and presents the localisation component interoperability environment developed as part of this research; Section 4 presents the architecture of LocConnect in detail; and section 5 discusses future work. The paper concludes with a summary of the present work and the contributions made by this study.

2. Background

Currently, software applications are increasingly moving towards a distributed model. Standards are vital for the interoperability of these distributed software applications. However, one of the major problems preventing successful interoperability between and integration of distributed applications and processes is the lack of (standardised) interfaces between them.

In order to address these issues, workflow interoperability standards have been proposed (Hayes et al 2000) to promote greater efficiency and to reduce cost. The Wf-XML message set defined by the Workflow Management Coalition (WfMC) and The Simple Workflow Access Protocol (SWAP) are examples of such internet-scale workflow standards (Hayes et al 2000). Most of these standards only define the data and metadata structure while standards such as Hyper-Text Transfer Protocol (HTTP), Common Object Request Broker Architecture (CORBA), and the Internet Inter-ORB Protocol (IIOP) focus on the transportation of data structures (Hayes et al 2000).

From a purely functional standpoint, we also have the Web Service Description Language (WSDL), the most recent version being WSDL 2.0 (W3C 2007). WSDL is an XML-based language that defines services as a collection of network endpoints or ports. It is regarded as being a simple interface definition language (Bichler and Lin 2006) which does not specify message sequence or its constraints on parameters (Halle et al 2010). However, while it does describe the public interface to a web service, it possesses limited descriptive ability and covers only the functional requirements in a machine-readable format. Where this becomes an issue is in defining a non-static workflow, as the interface does not provide enough information to allow a broker to make a value judgement in terms of other qualities that are of considerable interest in the localisation process, such as the quality, quantity, time and cost aspects discussed earlier. These service attributes are much more difficult to define, as they cover the non-functional aspects of a service, e.g. how well it is performed. This contrasts with the more Boolean functional requirements (either it complies with the service support requirements, or it does not). Therefore, WSDL does not provide sufficient coverage to support our requirements for interoperability.

There are some notable examples of localisation and translation-centric web services, such as those currently offered by Google, Bing and Yahoo!. However, even here we run into interoperability issues as the interfaces provided do not follow any specific standard, and connecting to these services is still very much a manual process requiring the intervention of a skilled computer programmer to set up the call to the service, to validate the data sent in terms of string length, language pair, and so on, and then to handle the data that is returned. Some localisation Translation Management Systems (TMS) purport to provide such flexibility, but they tend to be monolithic in their approach, using pre-defined workflows, and requiring dedicated developers to incorporate services from other vendors into these workflows through the development of bespoke APIs. What is needed is a unified approach for integrating components, so that any service can be called in any order in an automated manner.

2.1 The XLIFF Standard

The XML-based Localization Interchange File Format (XLIFF) is an open standard for exchanging localisation data and metadata. It has been developed to address various issues related to the exchange of localisation data.

The XLIFF standard was first developed in 2001 by a technical committee formed by representatives of a group of companies, including Oracle, Novell, IBM/Lotus, Sun, Alchemy Software, Berlitz, Moravia-IT, and ENLASO Corporation (formerly the RWS Group). In 2002, the XLIFF specification was formally published by the Organization for the Advancement of Structured Information Standards (OASIS) (XLIFF-TC 2008).

The purpose of XLIFF as described by OASIS is to "store localizable data and carry it from one step of the localization process to the other, while allowing interoperability between tools" (XLIFF-TC 2008). By using this standard, localisation data can be exchanged between different companies, organizations, individuals or tools. Various file formats such as plain text, MS Word, DocBook, HTML, XML etc. can be transformed into XLIFF, enabling translators to isolate the text to be translated from the layout and formatting of the original file format.

The XLIFF standard aims to (Corrigan & Foster 2003):

- Separate translatable text from layout and formatting data;
- Enable multiple tools to work on source strings;
- Store metadata that is helpful in the translation/localisation process.

The XLIFF standard is becoming the de facto standard for exchanging localisation data. It is accepted by almost all localisation service providers and is supported by the majority of localisation tools and CAT tools. The XLIFF standard is being continuously developed further by the OASIS XLIFF Technical Committee (2010).

2.2 Localisation Standards and Interoperability Issues

Although the adoption of localisation standards would very likely provide benefits relating to reusability, accessibility, interoperability, and reduced cost, software publishers often refrain from the full implementation of a standard or do not carry out rigorous standard conformance testing. There is still a perceived lack of evidence for improved outcomes and an associated fear of the high costs of standard implementation and maintenance. One of the biggest problems with regards to tools and technologies today is the pair-wise product drift (Kindrick et al 1996), i.e. the need for the output of one tool to be transformed in order to compensate for another tool's non-conforming behaviour. This trait is present within the localisation software industry. Although the successful integration of different software brings enormous benefits, it is still a very arduous task.

Most current CAT tools, while accepting and delivering a range of file formats, maintain their own proprietary data formats within the boundary of the application. This makes sharing of data between tools from different software developers very difficult, as conversion between formats often leads to data loss.

XLIFF, as mentioned above, intends to provide a solution to these problems, but true interoperability can only be achieved once the XLIFF standard is implemented in full by the majority of localisation tools providers. Currently, XLIFF compliance seems to be regarded as an addition to the function list of many localisation applications, rather than being used to the full extent of its abilities, and indeed many CAT tools seem to pay mere lip service to the XLIFF specification (Anastasiou and Morado-Vazquez 2010; Bly 2010), outputting just a minor subset of the data contained in their proprietary formats as XLIFF to ensure conformance.

3. Experimental Setup

With advancements in technology, the localisation process of the future can be driven by a successful integration of distributed heterogeneous software components. In this scenario, the components are dynamically integrated and orchestrated depending on the available resources to provide the best possible solution for a given localisation project. However, such an ideal component-based interoperability scenario in localisation is still far from reality. Therefore, in this research, we aim to model this ideal scenario by implementing a series of prototypes. As the initial step, an experimental setup has been designed containing the essential components.

The experimental setup includes multiple interacting components. Firstly, a user creates a localisation project by submitting a source file and supplying some parameters through a user interface component. Next, the data captured by this component is sent to a Workflow Recommender component. The Workflow Recommender implements the appropriate business process. By analysing source file content,

resource files as well as parameters provided by the user, the Workflow Recommender offers an optimum workflow for this particular localisation project. Then, a Mapper component analyses this workflow and picks the most suitable components to carry out the tasks specified in the workflow. These components can be web services such as Machine Translation systems, Translation Memory Systems, Post Editing systems etc. The Mapper will establish links with the selected components. Then a data container will be circulated among the different components according to the workflow established earlier. As this data container moves through different components, the components modify the data. At the end of the project's life cycle, a Converter component transforms this data container to a translated or localised file which is returned to the user.

Service Oriented Architecture is a key technology that has been widely adopted for integrating such highly dynamic distributed components. Our research revealed that the incorporation of an orchestration engine is essential to realise a successful SOA-based solution for coordinating localisation components. Furthermore, the necessity of a common data layer that will enable the communication between components became evident. Thus, in order to manage the processes as well as data, we incorporated an orchestration engine into the aforementioned experimental setup. This experimental setup along with the orchestration engine provide an ideal framework for the investigation of interoperability issues among localisation components.

3.1 LocConnect

At the core of the experimental setup are the orchestration engine and the common data layer, which jointly provide the basis for the exploration of interoperability issues among components. This prototype environment is called LocConnect. The following sections introduce the features of LocConnect and describe its architecture.

3.1.1 Features of LocConnect

LocConnect interconnects localisation components by providing access to an XLIFF-based data layer through an Application Programming Interface (API). By using this common data layer we allow for the traversal of XLIFF-based data between different localisation components. Key features of the LocConnect testing environment are summarized below.

- Common Data Layer and Application Programming Interface

LocConnect implements a common XLIFF-based datastore (see section 4.5) corresponding to individual localisation projects. The components can access this datastore through a simple API. Furthermore, the common datastore can also hold various supplementary resource files related to a localisation project (see section 4.4). Components can manipulate these resource files through the API.

- Workflow Engine

The orchestration of components is achieved via an integrated workflow engine that executes a localisation workflow generated by another component.

- Live User Interface (UI)

One of the important aspects of a distributed processing scenario is the ability to track progress along the different components. An AJAX-powered UI has been developed to display the status of the components in real-time. LocConnect's UI has been developed in a manner that allows it to be easily localised into other languages.

- Built-in post-editing component (XLIFF editor)

In the present architecture, localisation project creation and completion happens within LocConnect. Therefore, an online XLIFF editor was developed and incorporated into LocConnect in order to facilitate post-editing of content.

- Component Simulator

In the current experimental setup, only a small number of components, most of them developed as part of the CNGL research at the University of Limerick and other participating research groups, have been connected up. The Workflow Recommender, Mapper, Leveraging Component and a Translating Rating component are among these components. A component simulator was, therefore, developed to allow for further testing of interoperability issues in an automated localisation workflow using the LocConnect framework.

A single-click installer and administrator configuration panel for LocConnect were developed as a part of this work to allow for easy installation

and user-friendly administration.

3.1.2 Business Case

Cloud-based storage and applications are becoming increasingly popular. While the LocConnect environment supports the adhoc connection of localisation components, it can also serve as cloud-based storage for localisation projects. These and other key advantages of LocConnect from a business point of view are highlighted below.

- Cloud-based XLIFF and resource file storage

LocConnect can simply be used as a cloud-based XLIFF storage. Moreover, due to its ability to store resource files (e.g. TMX, SRX etc.), it can be used as a repository for localisation project files. As such, LocConnect offers a central localisation data repository which is easy to backup and maintain.

- Concurrent Versioning System (CVS)

During a project's life cycle, the associated XLIFF data container continuously changes as it travels through different localisation components. LocConnect keeps track of these changes and stores different versions of the XLIFF data container. Therefore, LocConnect acts as a CVS system for localisation projects. LocConnect provides the facility to view both data and metadata associated with the data container at different stages of a workflow.

- In-built Online XLIFF editor

Using the inbuilt online XLIFF editor, users can edit XLIFF content easily. The AJAX-based UI allows easy inline editing of content. Furthermore, the online editor shows alternative translations as well as useful metadata associated with each translation unit.

- Access via internet or intranet

With its single click installer, it can easily be deployed via the internet or an intranet. LocConnect can also act as a gateway application where LocConnect is connected to the internet while the components can safely reside within an intranet.

- Enhanced revenues

The LocConnect-centric architecture increases data exchange efficiency as well as automation. Due to increased automation, we would expect lower localisation costs and increased productivity.

3.2 Description of Operation (Use Case)

The following scenario provides a typical use case for LocConnect in the above experimental setup.

A project manager logs into the LocConnect server and creates a LocConnect project (a.k.a. a job) by entering some parameters. Then the project manager uploads a source file. The LocConnect server will generate an XLIFF file and assign a unique ID to this job. Next, it will store the parameters captured through its interface in the XLIFF file and embed the uploaded file in the same XLIFF file as an internal file reference. The Workflow Recommender will then pick up the job from LocConnect (see the procedure described in section 4.2.1), retrieve the corresponding XLIFF file and analyse it. The Workflow Recommender will generate an optimum workflow to process the XLIFF file. The workflow describes the other components that this XLIFF file has to go through and the sequence of these components. The Workflow Recommender embeds this workflow information in the XLIFF file. Once the workflow information is attached, the file will be returned to the LocConnect server. When LocConnect receives the file from the Workflow Recommender, it decodes the workflow information found in the XLIFF file and initiates the rest of the activities in the workflow. Usually, the next activity will be to send the XLIFF file to a Mapper Component which is responsible for selecting the best web services, components etc. for processing the XLIFF file. LocConnect will establish communication with the other specified components according to the workflow and component descriptions. As such, the workflow will be enacted by the LocConnect workflow engine. Once the XLIFF file is fully processed, XLIFF content can be edited online using LocConnect's built-in editing component. During the project's lifecycle, the project manager can check the status of the components using LocConnect's live project tracking interface. Finally, the project manager can download the processed XLIFF and the localised files.

4. Architecture

This section describes the LocConnect architecture in detail.

LocConnect is a web-based, client-server system. The design is based on a three-tier architecture as depicted in figure 1. The implementation of the

system is based on PHP and AJAX technologies.

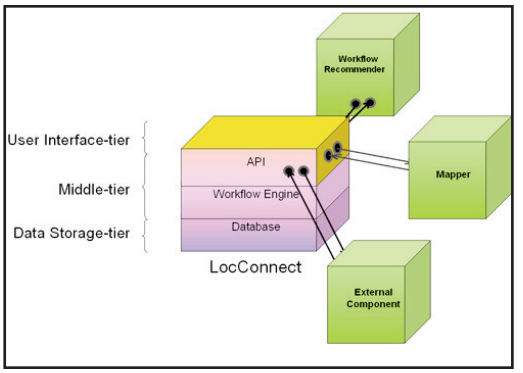

Figure 1. Three-tier architecture of LocConnect

User interface tier - a client-based graphical user interface that runs on a standard web browser. The user interface provides facilities for project management, administration and tracking.

Middle tier - contains most of the logic and facilitates communication between the tiers. The middle tier mainly consists of a workflow engine and provides an open API with a common set of rules that define the connectivity of components and their input output (IO) operations. The components simply deal with this interface in the middle tier.

Data Storage tier - uses a relational database for the storage and searching of XLIFF and other resource data. The same database is used to store information about individual projects.

The tiers are described below.

4.1 User Interface

Web-based graphical user interfaces were developed for:

1. Capturing project parameters during project creation;
2. Tracking projects (i.e. to display the current status of projects);
3. Post-editing translations;
4. Configuring the server and localising the interface of LocConnect.

During project creation, a web-based form is presented to a user. This form contains fields that are required by the Workflow Recommender to generate a workflow. Parameters entered through this interface will be stored in the XLIFF file along with the uploaded source file (or source text) and resource files. The project is assigned a unique ID through this interface and this ID is used throughout the project's lifecycle.

The project-tracking interface reflects the project's workflow. It shows the current status of a project, i.e. pending, processing, or complete in relation to each component. It displays any feedback messages (such as errors, warnings etc.) from components. The current workflow is shown in a graphical representation. Another important feature is a log of activities for the project. Changes to the XLIFF file (i.e. changes of metadata) during different stages of the workflow can be tracked. The project-tracking interface uses AJAX technologies to dynamically update its content frequently (see figure 2).

Figure 2. Project Tracking UI

At the end of a project's lifecycle, the user is given the option to post-edit its content using the built-in XLIFF post-editor interface. It displays source strings, translations, alternative translations and associated metadata. Translations can be edited through this interface. The Post-editing component also uses AJAX to update XLIFF files in the main datastore (see section 4.4). See figure 3 for a screen-shot of the post-editing interface. A preliminary target file preview mechanism has been developed and integrated into the same UI.

Figure 3. Post-Editing Interface

A password-protected interface has been provided for the configuration of the LocConnect server. Through this interface various configuration options such as LocConnect database path, component descriptions etc. can be edited. The same interface can be used to localise the LocConnect server itself (see figure 4 for a screenshot of the administrator's interface).

Figure 4. Administrator's Interface

The user interfaces were implemented in PHP, Javascript, XHTML and use the JQuery library for graphical effects and dynamic content updates.

4.2 Middle tier: Application Programming Interface (API)

The LocConnect server implements a Representational State Transfer (REST) - based interface (Fielding 2000) to send and retrieve resources, localisation data and metadata between components through HTTP-GET and HTTP-POST operations using proper Uniform Resource Identifiers (URI). These resources include:

- Localisation projects;
- XLIFF files;
- Resource files (i.e. files such as TBX, TMX, SRX etc.);
- Resource Metadata (metadata to describe resource file content).

The LocConnect API provides functions for the following tasks:

1. Retrieving a list of jobs pending for a particular component (list_jobs method);
2. Retrieving an XLIFF file corresponding to a particular job (get_job method);
3. Setting the status of a job. The status can be one of the following: Pending, Processing, Complete (set_status method);
4. Sending a feedback message to the server (send_feedback method);
5. Sending processed XLIFF files to the sever (send_output method);
6. Sending a resource file (i.e. a non-XLIFF asset file) to the server (send_resource method);
7. Retrieving a resource file from the server (get_resource method);
8. Retrieving metadata associated with a resource file (get_metadata method).

A complete description of each REST-based function is provided below.

Obtaining available jobs: list_jobs method

This method takes a single argument: component ID. It will return an XML containing the IDs of jobs pending for any given component. The IDs are alphanumeric and consist of 10 characters. The component ID is a string (usually, a short form of a component's name, such as WFR for Workflow

Recommender).

This method uses the HTTP GET method to communicate with the LocConnect server.

```
<jobs>
  <job>16674f2698</job>
  <job>633612fb37</job>
</jobs>
```

Retrieving the XLIFF file corresponding to a particular job: get_job method
This method takes two arguments: component ID and job ID. It will return a file corresponding to the given job ID and component ID. Usually, the file is an XLIFF file, however it can be any text-based file. Therefore, the returned content is always enclosed within special XML mark-up: <content>..</content>. The XML declaration of the returned file will be omitted in the output (i.e. <?xml version="1.0" ..?> will be stripped off from the output).

This method uses the HTTP GET method to communicate with the LocConnect server.

```
<content><xliffversion='1.2'xmlns='urn:oasis:names
:tc:xliff:document:1.2'>
<file original='hello.txt' source-language='en' target-
language='fr'  datatype='plaintext'>
  <body>
  <trans-unit id='hi'>
          <source>Hello world</source>
          <target>Bonjour le monde</target>
   </trans-unit>
   </body>
</file>
</xliff>
</content>
```

Setting current status: set_status method
This method takes three arguments: component ID, job ID, status. The status can be 'pending', 'processing' or 'complete'. Initially, the status of a job is set to 'pending' by the LocConnect server to mark that a job is available for pick up by a certain component. Once the job is picked by the component, it will change the status of the job to 'processing'. This ensures that the same job will not be re-allocated to the component. Once the status of a job is set to 'complete', LocConnect will perform the next action specified in the workflow.

This method uses the HTTP GET method to communicate with the LocConnect server.

Sending feedback message: send_feedback method
This method takes three arguments: component ID, job ID, feedback message. Components can send various messages (e.g. error messages, notifications etc.) to the server through this method. These messages will be instantly displayed in the relevant job tracking page of the LocConnect interface. The last feedback message sent to the LocConnect server before sending the output file will be stored within the LocConnect server and it will appear in the activity log of the job. The messages are restricted to 256 words in length.

This method uses the HTTP GET method to communicate with the LocConnect server.

Sending a processed XLIFF file: send_output method
This method takes three arguments: component ID, job ID and content. The content is usually a processed XLIFF file. Once the content is received by LocConnect, it will be stored within the LocConnect datastore. LocConnect will wait for the component to set the status of the job to 'complete' and move on to the next step of the workflow.

This method uses the HTTP POST method to communicate with the LocConnect server.

Storing a resource file: send_resource method
This method takes one optional argument: resource ID and two mandatory arguments: resource file and metadata description. The resource file should be in text format. Metadata has to be specified using the following notation:

Metadata notation: 'key1:value1-key2:value2-key3:value3'
e.g. 'language:en-domain:health'

If the optional argument resource ID is not given, LocConnect will generate an ID and assign that ID to the resource file. If the resource ID is given, it will overwrite the current resource file and metadata with the new resource file and metadata.

This method usew the HTTP POST method to communicate with the LocConnect server.

Retrieving a stored resource file: get_resource method
This method takes one argument: resource ID. Given the resource ID, the LocConnect server will return

the resource associated with the given ID.

This method uses the HTTP GET method to communicate with the LocConnect server.

Retrieving metadata associated with a resource file: get_metadata method
This method takes one argument: resource ID. The LocConnect server will return the metadata associated with the given resource ID as shown in the example below:

```
<metadata>
   <meta key="language" value="en">
   <meta key="domain" value="health">
</metadata>
```

This method uses the HTTP GET method to communicate with the LocConnect server.

4.2.1 Component-Server Communication Process
A typical LocConnect component-server communication process includes the following phases.

Step 1: list_jobs

This component calls the list_jobs method to retrieve a list of available jobs for that component by specifying its ID.

Step 2: get_job

This component uses get_job to retrieve the XLIFF file corresponding to the given job ID and the component ID.

A component may either process one job at a time or many jobs at once. However, the get_job method is only capable of returning a single XLIFF file at a time.

Step 3: set_status - Set status to processing

This component sets the status of the selected job to 'processing'.

Step 4: Process file

This component processes the retrieved XLIFF file. It may send feedback messages to the server while processing the XLIFF file. These feedback messages will be displayed in the job tracking interface of the LocConnect.

Step 5: send_output

This component sends the processed XLIFF file back to the LocConnect server using send_output method.

Step 6: set_status

This component sets the status of the selected job to 'complete'. This will trigger the LocConnect server to move to the next stage of the workflow.

4.3 Middle tier: Workflow Engine
A simple workflow engine has been developed and incorporated into the LocConnect server to allow for the management and monitoring of individual localisation jobs. The current workflow engine does not support parallel processes or branching. However, it allows the same component to be used several times in a workflow. The engine parses the workflow information found in the XLIFF data container (see section 4.5) and stores the workflow information in the project management datastore. The project management datastore is then used to keep track of individual projects. In the current setup, setting the status of a component to 'complete' will trigger the next action of the workflow.

4.4 LocConnect Datastore
The database design can be logically stratified in 3 layers:

- Main datastore holds XLIFF files;
- Project management datastore holds data about individual projects and their status;
- Resource datastore holds data and metadata about other resource files;

The main datastore is used to store XLIFF files corresponding to different jobs. It stores different versions of the XLIFF file that correspond to a particular job. Therefore, the LocConnect server also acts as a Concurrent Versions System (CVS) for localisation projects.

The project management datastore is used for storing the information necessary to keep track of individual localisation jobs with respect to localisation workflows. Furthermore, it is used to store various time-stamps such as job pick-up time, job completion time etc by different components.

The resource datastore is used to store various asset files associated with localisation projects. The asset files can be of any text-based file format such as TMX, XLIFF, SRX, TBX, XML etc. The components can store any intermediate files, temporary or backup files in this datastore. The files can then be accessed at any stage during workflow execution. The resource files (i.e. asset files) can be described further using metadata. The metadata consists of key-value pairs associated with the resource files and can also be stored in the resource datastore.

SQLite was chosen as the default database for implementing the logical data structure in this prototype, for a number of reasons. Firstly, it can be easily deployed. It is lightweight and virtually no administration required. Furthermore, it does not require any configuration.

4.5 XLIFF Data Container

The core of this architecture is the XLIFF-based data container defined in this research. Maximum effort has been made to abstain from custom extensions in defining this data container. Different components will access and make changes to this data container as it travels through different components and different phases of the workflow. The typical structure of the data container is given in figure 5.

When a new project is created in LocConnect, it will append parameters captured via the project creation page into the metadata section (see section 2) of the data container. The metadata is stored as key-value pairs. During the workflow execution process, various components may use, append or change the metadata. The source file uploaded by the user will be stored within the XLIFF data container as an internal file reference (see section 1). Any resource files uploaded during the project creation will also be stored as external-references as shown in section 4.4. The resource files attached to this data container can be identified by their unique IDs and can be retrieved at any stage during the process. Furthermore, the identifier will allow retrieval of the metadata associated with those resources.

After project creation, the data container generated (i.e. the XLIFF file) is sent to the Workflow Recommender component. It analyses the project metadata as well as the original file format to recommend the optimum workflow to process the given source file. If the original file is in a format other than XLIFF, the Workflow Recommender will suggest that the data container to be sent to a File Format Converter component. The file format converter will read the original file from the above internal-file reference and convert the source file into XLIFF. The converted content will be stored in the same data container using the <body> section and the skeleton sections. The data container with the converted file content is then reanalysed by the Workflow Recommender component in order to propose the rest of the workflow. The workflow information will be stored in section 3 of the data container. When the LocConnect server receives the data container back from the Workflow Recommender component, it will parse the workflow description and execute the rest of the sequence. Once the entire process is completed, the converter can use the data container to build the target file.

In this architecture, a single XLIFF-based data container is being used throughout the process. Different workflow phases and associated tools can be identified by the standard XLIFF elements such as <phase> and <tools>. Furthermore, tools can include various statistics (e.g. <count-groups>) in the same XLIFF file.

The XLIFF data container based architecture resembles the Transmission Control Protocol and the Internet Protocol (TCP/IP) architecture in that the data packet is routed based on its content. However, in this scenario, LocConnect plays several roles, including the role of a router, web server and a file server.

```
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
 <file original="hello.txt" source-language="en" target-language="fr" datatype="plaintext"
  category="medical">
  <header>

   <skl> <internal-file> </internal-file>  </skl>
```

Section 1

```
<reference>
 <internal-file form="base64">
 <original-file fileformat="exe"> </original-file>
 </internal-file>
</reference>
```

Section 2

```
<reference>
 <internal-file>
  <metadata>
   <meta pname="testProject"
    pdescription="A test project"
    startdate="01/04/2011"
    deadline="10/12/2011"
    budget="13310"
    quality-requirement="High"
    use-mt="yes"
    use-rating="yes"
   />
  </metadata>
 </internal-file>
</reference>
```

Section 3

```
<reference>
    <workflow>
        <task  tool-id="LMC" order="1" status="pending"/>
    </workflow>
</reference>
```

Section 4

```
    <reference>
      <external-file href="http:// LocConnect/get_resource.php?id=fcb4c5a8f1"/>
    </reference>

    <phase-group>
        <phase phase-name="Project Initiate" process-name="project creation and requirement
    capturing" tool-id="LocConnect"     company-name="LRC"/>

        <phase phase-name="Quality Assurance" process-name="authoring" tool-id="LKR"  company-
    name="LKR" contact-name="Dave O Carroll" contact-email="contact@example.com"/>
    </phase-group>

    <tool tool-name="LocConnect"  tool-id="LocConnect"  tool-version="2.0"/>

  </header>
  <body>..
  </body>
 </file>
</xliff>
```

Figure 5. XLIFF-Based Data Container

5. Discussion and Future Work

Savourel (2007) highlights the importance of a "Translation Resource Access API" which facilitates localisation data exchange among different systems in a heterogeneous environment. Like Savrourel (2007) we also believe that access to a common data layer through an API would enable interoperability between different localisation components. The development of the prototype has revealed syntactical requirements of such an API as well as the common data layer. Whilst the prototype provides a test bed for the exploration of interoperability issues among localisation tools, it has a number of limitations.

In the present architecture, metadata is being stored as attribute-value pairs within an internal file reference of the XLIFF data container (see section 3 of figure 5). However, according to the current XLIFF specification (XLIFF-TC 2008), XML elements cannot be included within an internal file reference. Doing so will result in an invalid XLIFF file. While this could be interpreted as a limitation of the XLIFF standard itself, the current metadata representation mechanism also presents several problems. The metadata is exposed to all the components. Yet there might be situations where metadata should only be exposed to certain components. Therefore, some security and visibility mechanisms have to be implemented for the metadata. Moreover, there may be situations where components need to be granted specific permissions to access metadata, e.g. read or write. These problems can be overcome by separating the metadata from the XLIFF data container. That is, the metadata has to be stored in a separate datastore (as in the case of resource files). Then, specific API functions can be implemented to manipulate metadata (e.g. add, delete, modify, retrieve) by different components. This provides a secure mechanism to manage metadata.

The Resource Description Framework (RDF) is a framework for describing metadata (Anastasiou 2011). Therefore, it is worthwhile exploring the possibility of representing metadata using RDF. For example, API functions could be implemented to return the metadata required by a component in RDF syntax.

The current API lacks several important functions. Functions should be implemented for deleting projects (and associated XLIFF files), modifying projects, deleting resource files and modifying metadata associated with resource files etc. The current API calls set_output and set_status to 'complete' could be merged (i.e. sending the output by a component will automatically set its status to 'complete'). Furthermore, a mechanism could be implemented for granting proper permissions to components for using the above functions. User management is a significant aspect that we did not pay much attention to when developing the initial test bed. User roles could be designed and implemented so that users with different privileges can assign different permissions to components as well as different activities managed through the LocConnect server. This way, data security could be achieved to a certain extent. Furthermore, an API key should be introduced for the validation of components as another security measure. This way, components would have to specify the key whenever they use LocConnect API functions in order to access the LocConnect data.

The XLIFF data container could contain sensitive data (i.e. source content, translations or metadata) which some components should not be able to access. A mechanism could be implemented to secure the content and to grant permissions to components so that they would only be able to access relevant data from the XLIFF data container. There are three potential solutions to this problem. One would be to let the workflow recommender (or the Mapper) select only secure and reliable components. The second solution could be to encrypt content within the XLIFF data container. The third solution could be to implement API functions to access specific parts of the XLIFF data container. However, the latter mechanism will obviously increase the complexity of the overall communication process due to frequent API calls to the LocConnect server.

Because the XLIFF standard was originally defined as a localisation data exchange format, it has, so far, not been thoroughly assessed with regard to its suitability as a localisation data storage format or as a data container. A systematic evaluation has to be performed on the use of XLIFF as a data container in the context of a full localisation project life cycle, as facilitated by our prototype. For example, during the traversal, an XLIFF-based data container could become cumbersome causing performance difficulties. Different approaches to addressing likely performance issues could be explored, such as data container compression, support for parallel processing, or the use of multiple XLIFF-based data

containers transmitted in a single compressed container. The implications of such strategies would have to be evaluated, such as the need to equip the components with a module to extract and compress the data container.

While the current workflow engine provides essential process management operations, it currently lacks more complex features such as parallel processes and branching. Therefore, incorporation of a fully-fledged workflow engine into the LocConnect server is desirable. Ideally, the workflow engine should support standard workflow description languages such as Business Process Execution Language (BPEL) or Yet Another Workflow Language (YAWL). This would allow the LocConnect server to be easily connected to an existing business process, i.e. localisation could be included as a part of an existing workflow. In the current system, the workflow data is included as an internal file reference in the XLIFF data container (see section 3 of figure 5) which invalidates the XLIFF file due to the use of XML elements inside the internal file reference. In future versions, this problem can be easily addressed by simply storing the generated workflow as a separate resource file (e.g. using BPEL) and providing the link to the resource file in the XLIFF data container as an external file reference.

LocConnect implements REST-based services for communication with external components. Therefore, it is essential to implement our own security measures in the REST-based API. Since there are no security measures implemented in the current LocConnect API, well-established and powerful security measures such as XML encryption, API keys would need to be implemented in the API as well as in the data transmission channel (e.g. the use of Secure Socket Layer (SSL) tunnels for REST calls).

Currently, the LocConnect server implements a 'PULL' based architecture where components have to initiate the data communication process. For example, components must keep checking for new jobs in the LocConnect server and fetch jobs from the server. The implementation of both 'PUSH' and 'PULL' based architectures would very likely yield more benefits. Such architecture would help to minimize communication overhead as well as resource consumption (e.g. the LocConnect server can push a job whenever a job is available for a component, rather than a component continuously checking the LocConnect server for jobs). The implementation of both 'PUSH' and 'PULL' based architectures would also help to establish the availability of the components prior to assigning a job, and help the LocConnect server to detect component failures. The current architecture lacks this capability of identifying communication failures associated with components. If the LocConnect server could detect communication failures, it could then select substitute components (instead of failed components) to enact a workflow. An architecture similar to internet protocol could be implemented with the help of a Mapper component. For example, whenever the LocConnect server detects a component failure, the data container could be automatically re-routed to another component that can undertake the same task so that the failure of a component will not affect the rest of the workflow.

The current resource datastore is only capable of storing textual data. Therefore, it could be enhanced to store binary data too. This would enable the storing of various file formats including windows executable files, dll files, video files, images etc. Once the resource datastore is improved to store binary data, the original file can be stored in the resource datastore and in XLIFF, and a reference to this resource can be included as an external file reference (see section 1 of figure 5).

In the present architecture, the information about components has to be manually registered with the LocConnect server using its administrator interface. However, the architecture should be improved to discover and register ad-hoc components automatically.

5.1 Proposed improvements to the XLIFF based data container and new architecture

By addressing the issues related to the above XLIFF-based data container, a fully XLIFF compliant data container could be developed to evaluate its effect on improvements in interoperability. A sample XLIFF data container is introduced in figure 6.

This data container differs from the current data container (see figure 5) in the following aspects:

The new container:

- Does not represent additional metadata (i.e. metadata other than that defined in the XLIFF specification) within the data container itself. Instead, this metadata will be stored in a separate metadata store that can be accessed via

corresponding API functions.

- Does not represent workflow metadata as an internal file reference. Instead, the workflow metadata will be stored separately in the resource datastore. A link to this workflow will then be included in the XLIFF data container as an external file reference (see section 2 of figure 6).

- Does not store the original file as an internal file reference. It will also be stored separately in the resource datastore. An external file reference will be included in the XLIFF file as shown in section 1 of figure 6.

The new data container does not use any extensions to store additional metadata or data, nor does it use XML syntax within internal-file elements. Thus, the above architecture would provide a fully XLIFF compliant (i.e. XLIFF strict schema compatible) interoperability architecture. Due to the separation of the original file content, workflow information and metadata from the XLIFF data container, the container itself becomes lightweight and easy to manipulate. The development of a file format converter component based on this data container would also be uncomplicated.

6. Conclusions

In this paper we presented and discussed a service-oriented framework that was developed and then applied to evaluate interoperability in localisation process management using the XLIFF standard. The use cases, architecture and issues of this approach were discussed. A prototype of the framework was successfully demonstrated at the CNGL Public Showcase in Microsoft, Ireland, in November 2010.

The framework has revealed the additional metadata and related infrastructure services required for linking distributed localisation tools and services. It has also been immensely helpful in identifying prominent issues that need to be addressed when developing a commercial application.

The prototype framework described in this paper is the first to use XLIFF as a data container to address interoperability issues among localisation tools. In our opinion, the successful implementation of this pilot prototype framework suggests the suitability of XLIFF as a full project life-cycle data container that can be used to achieve interoperability in localisation processes. The development of the above prototype has mostly focused on addressing the syntactic interoperability issues in localisation processes. The future work will mainly focus on addressing the semantic interoperability issues of localisation processes by improving the proposed system. The LocConnect framework will serve as a platform for future research on interoperability issues in localisation.

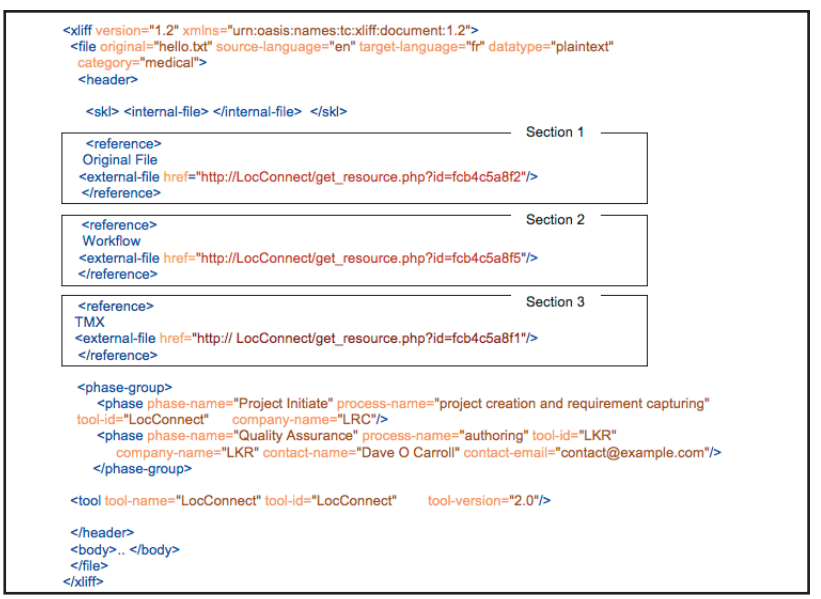

Figure 6. Improved Data Container

Acknowledgement

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at the Localisation Research Centre (Department of Computer Science and Information Systems), University of Limerick, Limerick, Ireland. We would also like to acknowledge the vital contributions of our colleagues and fellow researchers from the CNGL project.

References

Anastasiou, D. and Morado-Vazquez, L. (2010) 'Localisation Standards and Metadata', Proceedings Metadata and Semantic Research, 4th International Conference (MTSR 2010). Communications in Computer and Information Science, Springer, 255-276.

Anastasiou, D. (2011) 'The Impact of Localisation on Semantic Web Standards', in European Journal of ePractice, N. 12, March/April 2011, ISSN 1988-625X, 42-52.

Bichler, M. and Lin, K. J. (2006) 'Service-oriented computing'. IEEE Computer 39(3), 99-101.

Bly, M. (2010) 'XLIFFs in Theory and in Reality' [online], available: http://www.localisation.ie/xliff/resources/presentations/xliff_symposium_micahbly_20100922_clean.pdf [accessed 09 Jun 2011].

Corrigan, J. & Foster, T. (2003) 'XLIFF: An Aid to Localization' [online], available: http://developers.sun.com/dev/gadc/technicalpublications/articles/xliff.html [accessed 22 Jun 2009].

Fielding, R. (2000) 'Architectural Styles and the Design of Network-based Software Architectures' [PhD], University of California, Irvine.

Halle, S., Bultan, T., Hughes, G., Alkhalaf, M. and Villemaire, R. (2010) 'Runtime Verification of Web Service Interface Contracts', Computer, 43(3), 59-66.

Hayes, J. G., Peyrovian, E., Sarin, S., Schmidt, M. T., Swenson, K. D. and Weber, R. (2000) 'Workflow interoperability standards for the Internet', Internet Computing, IEEE, 4(3), 37-45.

IEEE. (1991) 'IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries', IEEE Std 610, 1.

Kindrick, J. D., Sauter, J. A. and Matthews, R. S. (1996) 'Improving conformance and interoperability testing', StandardView, 4(1), 61-68.

Lewis, G. A., Morris, E., Simanta, S. and Wrage, L. (2008) 'Why Standards Are Not Enough to Guarantee End-to-End Interoperability', in Proceedings of the Seventh International Conference on Composition-Based Software Systems (ICCBSS 2008), 1343630: IEEE Computer Society, 164-173.

Savourel, Y. (2007) 'CAT tools and standards: a brief summary', MultiLingual, September 2007, 37.

Schäler, R. (2009) 'Communication as a Key to Global Business'. In: Hayhoe, G. Connecting people with technology: issues in professional communication. Amityville N.Y. Baywood Pub. 57-67.

W3C. (2007) 'Web Service Description Language (WSDL)' Version 2.0 Part 1:Core Language, W3C Recommendation [online]. In Chinnici, R., Moreau, J, J., Ryman, A. and Weerawarana, S. eds.W3C, http://www.w3.org/TR/wsdl20.

XLIFF Technical Committee. (2008). 'XLIFF 1.2 Specification' [online]. http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html [accessed 25 Jun 2009].

XLIFF Technical Committee. (2010) 'XLIFF2.0 / Feature Tracking' [online], http://wiki.oasis-open.org/xliff/XLIFF2.0/FeatureTracking, [accessed 23 Jul 2009].

Localisation in International Large-scale Assessments of Competencies: Challenges and Solutions

Britta Upsing[1], Gabriele Gissler[1], Frank Goldhammer[1], Heiko Rölke[1], Andrea Ferrari[2]
[1]German Institute for International Educational Research,
Schloßstraße 29,
60486 Frankfurt am Main
www.tba.dipf.de
[2] cApStAn Linguistic Quality Control,
Chaussée de la Hulpe 268,
1170 Bruxelles,
www.capstan.be
upsing@dipf.de, gissler@dipf.de, goldhammer@dipf.de, roelke@dipf.de, andrea.ferrari@capstan.be

Abstract

International comparative studies like the Programme for International Student Assessment (PISA) pose special challenges to the localisation of the test content. To allow for comparison between countries, the assessments have to be comparable with respect to measurement properties. Therefore, internationalisation and localisation are crucial steps to guarantee test equivalence across countries. The localisation of test items is different from the localisation of web-based contents or software as the test content has to be authentic within a country while the test's measurement properties have to be comparable across countries.

Using the PIAAC study (Programme for the Assessment of Adult Competencies) as an example, this paper describes all stages of the localisation process for an international large-scale assessment. The process ranges from the development of source items to translation, adaptation of layout issues and meta-data adaptations. The paper concludes with a discussion of lessons learned and open questions.

1. Localisation in large-scale assessments

Most software or website localisation projects have the "ultimate aim of releasing a product that looks like it has been developed in country" (LISA 2003, p.11). This aim is reasonable for many instances of localisation. However, when moving to international large-scale assessment studies (studies that aim to compare skills or competence levels for given populations across countries, with a view to e.g. informing education policies), localisation is subjected to the primacy of comparability of assessment results, which may conflict with the aim of making a localised product look like it was developed in the target country itself. Unlike other localisation projects, localising assessments has to be undertaken with an eye on the comparability of multiple target versions of assessment instruments (e.g. tests). If translated tests behave differently in different countries (e.g. the difficulty varies across language versions), the significance of the research is at stake. This article will describe this potential conflict between authenticity and comparability when localising large-scale assessments on the basis of a case study.

In the remaining part of section 1, we will define large-scale assessments and add the most important details regarding the case study; this is followed by an overview of the particularities of localisation in large-scale assessment compared to web or software localisation processes. In Section 2, we will describe how these challenges can be met and show practical examples from our case. Section 3 will give an overview of the lessons learned.

1.1 What is large-scale assessment?

Policy makers around the globe need internationally comparable information about the outcomes of their education systems, information on what pupils know, and an overview of the skills and competencies of their adult workforce. This need has led to the introduction of international large-scale assessment studies, and since their implementation, localising the test content has become an important issue in the field.

In the current context, the term large-scale assessment (LSA) refers to national or international assessments that serve to describe population characteristics with respect to educational conditions

and learning outcomes, e.g. the competence level in a particular population. Basically, LSA studies are used for monitoring the achievement level in a particular population, for comparing assessed (sub)populations, and also for instructional programme evaluation. Such assessments may form the basis for developing and/or revising educational policies.

The International Association for the Evaluation of Educational Achievement (IEA) was one of the first organisations to implement international LSA studies to assess student achievement across countries. In 1995, IEA implemented TIMSS (Trends in International Mathematics and Science Study) to assess student achievement in mathematics, just to mention one example (Mullis et al. 2009). The most widely known LSA study is the Programme for International Student Assessment (PISA) by the Organisation for Economic Co-operation and Development (OECD). The first PISA cycle took place in 2000; cycles are repeated every three years. By 2012, more than 70 countries will have participated in PISA. PISA intends to measure the knowledge and skills of fifteen-year-old students and thus make inferences on the performance of the participating countries' education systems (OECD 2010). A very first step in the shift to computer-based assessment was made in 2006 when three countries took part in the computer-based assessment of science. In 2009, participating countries had the option to evaluate the digital reading skills of their students, and a more substantial shift to the computer-based test mode was taken. 19 countries opted for this assessment (OECD 2011).

There have also been several attempts to measure the competencies of adult populations (cf. Thorn 2009): In 1994, the OECD introduced the first cycle of the International Adult Literacy Survey (IALS) to obtain information about adult literacy (prose literacy, document literacy, and quantitative literacy) in participating countries and two more rounds followed (1996 and 1998). Altogether 22 countries participated in this survey. The OECD Adult Literacy and Lifeskills Survey (ALL) builds on the results of this study and provides an international comparison of literacy, numeracy and problem-solving skills in 12 countries. It took place between 2002 and 2006. This study is now followed by the Programme for the International Assessment of Adult Competencies (PIAAC), an international large-scale survey that assesses the skills of a representative sample of adults in 25 countries.

This paper will use the example of PIAAC to describe the localisation process in LSA studies. Like PISA, PIAAC is an OECD study. PIAAC is supposed to help governments to receive "high-quality comparative information regarding the fundamental skills of the adult population" (Schleicher 2008, p. 628). The target population consists of 16-65 year old adults. The project is run by an international consortium (that includes the authors of this paper) that is responsible for enabling the local project teams to conduct the study in their respective countries. The implementation of PIAAC started in 2007. The field study[1] took place in 2010; the main study will be carried out in 2011 and 2012. Results will be published in 2013. PIAAC-tests are subdivided into three different subject domains: literacy, numeracy and problem-solving in a technology-rich environment. In each of the domains, the competencies of the test participants are assessed by a number of test items[2] of varying difficulty.

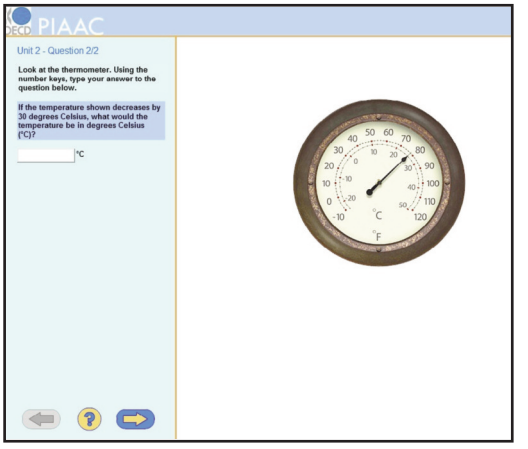

Figure 1: Sample numeracy test item (question on the left, stimulus material on the right)

The assessment items are preceded by a

[1] The field study serves to prepare the main study in several respects. Major goals are to evaluate the survey operations (e.g. sampling, data collection), and to investigate empirically the assessment instruments including their psychometric characteristics (e.g. comparability across countries). Based on field study results, data collection procedures and assessment instruments are revised (e.g. by dropping ill-fitting items).

[2] In our context, an item is the smallest assessable entity of a test. It consists of a stimulus that serves to evoke an observable response from the test taker; this is the material that the subject uses to answer the question. Individual differences in the response are assumed to reflect individual differences in the assessed ability or competence. Multiple items assessing the same ability form a test that allows to measure individual ability levels reliably. Individual response patterns observed across the items of a test are the empirical basis for estimating the subjects' ability levels. Multiple items including one shared stimulus are usually referred to as a unit.

questionnaire which collects background information about the test participant. The sample includes 5000 completed interviews per country. PIAAC is a household study: the interview and the test itself take place in a respondent's home (Thorn 2009). PIAAC is the first international LSA study that is completely computer-based[3], and therefore the first study to meet the specific challenges resulting from this test mode. Other studies are likely to follow this trend (e.g. PISA 2015).

As previously mentioned, localisation is an important issue because all assessment instruments (i.e., tests and questionnaires) have to be made available in the national language(s) of every participating country. PIAAC and other LSAs are challenged by localising the test items while maintaining the comparability of assessment results across countries and languages. This will be further elaborated in the next section.

1.2 Particularities of internationalisation and localisation in assessments

The localisation of LSA boils down to two questions: What exactly does it mean to internationalise and localise a test? How is this different from other localisation projects?

Adaptation of test items can occur in two scenarios and is not limited to large-scale assessments. In the first scenario, a test is originally developed for a specific language and its specific national context. Using the test internationally is not an issue when developing the test items. If, later on, the need arises to adapt the test for a new culture and language, the goal may be to obtain strict comparability, or the source test may just serve as the blueprint of a new test. This means that test developers have to decide "whether test adaptation is the best strategy" (Hambleton 2002, p. 65). In the second scenario, which is typical in the LSA context, the intended use of the test in an international comparison is a crucial factor right from the outset of developing the test. This is to ensure "that a person of the same ability will have the same probability of answering any assessment item successfully independent of his or her linguistic or cultural background" (Thorn 2009, p.8). Hence, in this second scenario, internationalisation plays an important role in making sure that the adaptation of the test will be feasible.

For computer-based tests, linguistic, cultural and technical aspects have to be taken into account to create "internationalised" source[4] items. The following definition by Schäler (2007, p.40) is applicable for the internationalisation of LSA studies as well:

> *"Internationalisation is the process of designing (or modifying) digital content (in its widest sense) . . . to isolate the linguistically and culturally dependent parts of an application and of developing a system that allows linguistic and cultural adaptation supporting users working in different languages and cultures."*

From a conceptual point of view, this means that source item content has to be created that is meaningful and authentic in all target cultures, as well as easily translatable. From a technical point of view, software developers have to make sure that translators can easily edit all adaptable content.

In a second step, the adaptable content has to be localised. Localisation is defined by Schäler (2007, p.40) as follows:

> *"Localisation is the linguistic and cultural adaptation of a digital product or service to the requirements of a foreign market and the management of multilinguality across the global, digital information flow."*

In the context of LSA, not all of these factors play an important role. While Schäler emphasizes the adaptation for the target culture and making sure that the product works in the target culture, in the context of LSA, it is important that test items remain comparable across different language versions. The creation of test items for an international comparative test is thus highly demanding. On the one hand, it is important that the items are authentic within a country; on the other hand, they have to be comparable across countries. This is one of the crucial aspects that differ from other localisation processes, resulting in a multi-step adaptation process.

A second aspect deals with the material that has to be localised. In a computer-based test like PIAAC,

[3] It should be noted though that there is a paper-based component for test participants that are not familiar with using a computer.

[4] "Source" and "target" are used in this paper in the usual meaning in the translation context: the source text (or in our case the source item) refers to all aspects of an item, i.e., text, graphic elements, scoring information etc., which are being translated and/or adapted. The target text (or target item) is the translated and adapted version of the source text (source item).

localisation is not limited to the content of a test item. Meta-data like material related to the correct and incorrect responses of test items will have to be adapted as well. This is an aspect that plays a key role in the localisation process of computer-based LSA. In computer-based tests this meta-data will have to be changed in the system itself to enable automatic scoring (detailed information on this process follows in section 2.2.2).

Section 2 will explain how these two aspects are tackled in the LSA study PIAAC.

2. Case study: Localising PIAAC assessment instruments

Section 1.2 showed that the context of LSA places special requirements on the localisation process. In PIAAC, this challenge was met by first internationalising and then localising the test content. Section 2.1 describes how this was done by first creating 'internationalised' source versions of test items, while section 2.2 contextualises the insights into the localisation process itself with a focus on quality assurance.

2.1 Internationalising test items

Before the item development process can start, the "competence" that shall be measured by these items has to be defined. Basically, a competence is a theoretical construct that is used to explain and predict individual differences in behavior. Most educational LSA studies target the assessment of individual differences in competencies like "reading literacy" (in broad terms: how well can the test participant read and understand text?) or "numeracy" (again in broad terms: how well can the test participant deal with mathematical demands?). Defining the construct is a complicated process and "construct equivalence in the languages and cultures of interest" has to be kept in mind (Hambleton 2002, p. 65). Once the construct is specified and refined by an international expert group, the experts derive an "assessment framework" on the basis of the construct definition (cf. Kirsch 2001). This assessment framework explains how the test and task characteristics are related to the construct definition, and it provides systematic information about the required combinations of task characteristics to cover the construct. The creation of items can start once the assessment framework is set. In all LSA studies mentioned in chapter 1.1, the source items (see Figure 1 for an example) are created in English. They form the basis for the later localisation process. Throughout the entire item development process, the international perspective takes an all-pervasive role and several qualitative control mechanisms are in place to make sure that linguistic and intercultural aspects are considered from as many linguistic and cultural perspectives as possible. A detailed description of how such a process can be established can be found in McQueen and Mendelovits (2003). When the source items are developed, the focus is already on authenticity and comparability. The processes involved in ensuring that authentic and comparable items are created will be explained in section 2.1.1 and 2.1.2.

2.1.1 Authenticity of item content

In most software or web localisation projects, authenticity is the "ultimate aim" (LISA 2003, p. 11) as the localised projects are supposed to look like they were developed in the target country itself. For LSA studies, this means that test items should be authentic. These items should represent demands that are common and typical within a country. Furthermore, items should include task requirements that are encountered by members of the target population in their daily life. Real-life scenarios, however, are different across countries: a Japanese scenario may not be authentic in Chile. For instance, an item that asks the test participant to do a Google search and to evaluate the search results may be very authentic in many countries, but it is unfamiliar to most Koreans (where the Google search engine is hardly used). The second goal in LSA studies, i.e. comparability between localised versions, might be compromised if an item's context is familiar to some countries' populations but completely unknown in others. All localised versions of an item should function like the source version of the item, thereby yielding a high level of psychometric comparability across localised versions. The major goal is that an item has the same degree of difficulty for all countries and measures the respective construct equally well across all countries.

Hence, when item developers create the source version of a test item, they try to look for the lowest common denominator. This holds the risk of creating item material that is "bland" because the common denominator is too low. As a compromise, the following approach as used for the PISA reading assessment may be reasonable:

> *"The aim (...) was not to produce an instrument whose content and contexts were completely familiar to all participating*

> *students, but, as far as possible, to control the occurrence of unfamiliarity so that no single cultural or linguistic group would be placed at a disadvantage." (McQueen and Mendelovits 2003, p. 216)*

Item developers thus need to be careful when their items refer to national aspects, e.g. certain locations, institutions, education systems, currencies etc., as this raises many questions: Is the aspect known in all participating countries? Does the level of familiarity have an impact on the difficulty of the task? Is this aspect a fundamental for covering the construct?

For example, items that include aspects concerning a particular national education system raise problems even if every country might be able to localise the provided information. Educational terms (e.g. community college) can have different meanings in different countries - and be completely unknown in others. Another issue that could make a test item less authentic in some countries is any reference to the climate or weather in relation to different seasons/months. Though a scenario involving a summer party taking place in July is realistic in Europe, this scenario is not plausible in Australia.

Decisions on how to ensure authenticity have to be made on a case-by-case basis and alternative solutions are possible. Item developers could decide to replace the national reference with a fictitious name, and consequently standardise the required level of the tested persons' ability to abstract (e.g. in PISA, zed is the fictional currency unit). If the source version is not standardised in this way, item developers have to indicate to translators how to deal with this issue (e.g. if standardisation is recommended, translators might be advised to "find an equivalent institution in your country" or if standardisation is not recommended, they might be asked to "use the existing name of the institution although this institution is unknown in your country"). In most LSA studies, item developers are supported by international content experts and the participating countries themselves in making these decisions and in selecting or designing suitable items (cf. McQueen and Mendelovits 2003).

In PIAAC, similar measures were taken to control the degree of unfamiliarity across countries. Domain expert groups were set up to represent a wide range of languages and cultures. These expert groups were responsible for creating the assessment framework, which served as a basis for creating items. The item developers created items that simulate authentic real-life scenarios. The experts checked these items keeping an eye on familiarity across cultures. The selected items were presented to representatives of the participating countries, who were given the opportunity to check early versions of the items for cultural bias. Only those items that were accepted by countries were translated and used for the field test. Following the field test, items that worked inconsistently across countries were dropped or modified before being included in the main study.

2.1.2 Further measures for enabling comparability

To avoid item translations that could jeopardise comparability between localised versions, several measures related to linguistic and layout issues can be implemented when preparing the source items:

1) Careful linguistic construction of the source text to ensure translatability

2) Guidelines informing translators about the degree to which they can adapt translations to their countries

3) Central control of the layout of the item

4) Control of adaptable parts of an item

To ensure translatability, item developers refer to a number of general guidelines. For example, they should only use idiomatic speech in the source version of an item with great care, as it could be very difficult to find adequate formulations for each of the target languages. Also, it might be difficult to find adequate translations of things like proverbs. Item questions should not be directed at the "level of nuances of language" (McQueen and Mendelovits 2003, p.215). Generally, the passive voice should be avoided because it does not exist in all languages (Hambleton 2002, p. 71).

Item creation must be accompanied by detailed translation guidelines for preparing the subsequent localisation process, otherwise comparability between target versions would be questionable from the outset because translators for different languages might assume different degrees of "translating freedom". These guidelines should answer all questions that a translator may have regarding the adaptation of specific item content ("Can I adapt the number format to the number format that is used in my country?", "Can I adapt the name of the institution?"…). In addition, guidelines should

provide general instructions for the translation of assessment items. This can include explaining which style of speech needs to be used in certain settings, general information about translating assessment items. For example: make sure that answer choices are kept about the same lengths in the translation so that they do not become a clue to the correct answer, information about the target audience, etc. (Hambleton, Merenda & Spielberger 2005).

In PIAAC, translators received "translation guidelines" with general instructions on how to translate assessment items. A second document, the "translation and adaptation guidelines", describes the structure and content of each item as well as the correct and incorrect answers. It gives advice for translating item-specific content, e.g. on how the translator should deal with names (adapt or not?). In addition to the general translation and adaptation guidelines, a so-called verification follow up form (VFF) is used to organise and control the localisation process. The VFF is a spreadsheet containing all text elements of an item and related instructions, including precise translation/adaptation advice relating to specific text elements (what should be adapted, what should not, how to understand ambiguous or difficult terms, pointers on consistency both within and across units, etc.). The VFF serves as a means to document all comments and successive translated versions of each item as it goes through the different phases of the localisation process: double translation and reconciliation, verification, country's post verification review, layout adaption, finalisation (for more details, see section 2.2.1).

The context of LSA studies may involve specific requirements regarding item layout when designing the source versions. Item developers want to be in control of the item layout across language versions as the position of information that is crucial for completing a task may affect item difficulty (Freedle 1997). This is the case when scrolling is required to see all of the text included in an item (for example in a stimulus that imitates a webpage); or when a long text is divided into several columns. To ensure comparability in these cases, it may be important that the starting position of text elements like headlines, paragraphs or the location of the correct response is exactly the same for all language versions. This could be solved by designing the source version in a way that precludes the introduction of cross-country variability in critical properties of the text layout. Therefore, the item editing software should allow for defining the absolute position of each element on the screen. In PIAAC, the CBA ItemBuilder was used as a tool for developing the source version of test items. The concept of the CBA ItemBuilder is to enable item writers to design and edit computer-based test items with the aid of a graphical editor that can easily be used by non-IT-specialists. The different components of an item can be positioned in the drawing area. The item writer has full control over the absolute size and position of the different components because each element can be aligned pixel by pixel on the screen. Consequently, the location of these elements cannot be changed when the text is translated. In anticipation of layout problems that could occur after translating the English source version to the different languages, the size of each text field was not only made as large as necessary for the English text, but was enlarged by approximately 1/3 to have enough space for languages that require more space for the same content, e.g. German or Russian.

Finally, with regard to the subsequent localisation process, it needs to be decided which components of the source items need to be adaptable, and which should be static across language versions. Basically, only those elements which are meant to be translated or adapted during the localisation process should be adaptable. Otherwise, comparability may be compromised due to uncontrolled changes.

An item usually consists of graphical and textual elements. For computer-based items, these textual elements can also include meta-data like scoring information. All textual elements need to be adaptable for translating the content to the target language. In addition, one could a[l]so think of adapting the graphical elements of a test item. For example, this would be necessary when adapting an item that simulates a website to a right-to-left written language system. To achieve an authentic context for this language versio[n], not only does the text need to be adapted, but also the text layout and the website structure.

In PIAAC, none of the participating countries used a right-to-left written language system; therefore only textual elements were made adaptable. Also, all countries were supposed to use the same images as the source item. As a consequence, textual and graphical elements needed to be technically separable. Moreover, graphics should not contain any textual elements but if needed were superimposed by textual elements. Even symbols were to be avoided or at least checked in terms of their international

suitability.

The software that was used for building the source versions allowed the separation of the entire textual content from the graphical representation of an item, and to export this adaptable content as an XLIFF5 file. Later on in the localisation process, this XLIFF file was used for translation purposes. Once the text had been translated and validated, the XLIFF file was reimported to the test item.

The finalised internationalisation process results in a set of carefully checked and reviewed source items. These items serve as a basis for the localisation process, which will be described in section 2.2.

2.2 Localising test items

The localisation process consists of several steps to obtain items that can function comparably across countries as well as being authentic within a country. The content - mostly text - included in the item has to be adapted, but in several cases the layout or the scoring has to be adapted as well.

Section 2.2.1 will describe the adaptation and quality assurance procedures involved in adapting the textual content, section 2.2.2 will describe the layout adaptations, and section 2.2.3 will explain why metadata such as the scoring of an item may have to be adapted as well, and how this can be done.

2.2.1 Localising the content

The International Test Commission Test Adaptation Guidelines (cf. Hambleton and de Jong 2003, p. 129) ask for a highly sophisticated translation procedure:

> "D.5 Test developers/publishers should implement systematic judgmental evidence, both linguistic and psychological, to improve the accuracy of the adaptation process and compile evidence on the equivalence of all language versions."

The translations should correctly deliver the content, be authentic and fluent, and at the same time they must not change the psychometric properties of the item. Thus, for LSA it is recommended to set up rigorous translation procedures that involve more than one translator for the adaptation of test items. Also, one individual can hardly meet the required translator's profile:

> "There is considerable evidence suggesting that test translators need to be (1) familiar with both source and target languages and the cultures, (2) generally familiar with the construct being assessed, and (3) familiar with the principles of good test development practices." (Hambleton 2002, p. 62)

For LSA, the double-translation design is recommended. Double-translation means that two translators create two independent translations of the source text. This is followed by a "reconciliation", which consists of merging the two independent translations into one target version. As Grisay (2003, p. 228) puts it:

> "equivalence of the source and target languages is obtained by using three different people (two translators and one reconciler) who all work on the (sic!) both source and the target versions."

In general, the idea is bringing together linguistic, domain and assessment experts that work as a team in creating the best possible target version.

In PIAAC, double-translation and reconciliation were carried out by the project teams within countries and the translation efforts were subsequently checked by a "verification" process provided by the international consortium in charge of the project. Specially recruited and trained verifiers checked both formal correspondence of target version to the source version and fluency/correctness in the target version, striving to achieve an optimal balance between these two goals, which sometimes pull different ways (e.g. maintaining the order of presentation of the information within a sentence or passage - versus opting for a more "natural" order in the target language). They also check whether the above-mentioned layout and adaption guidelines are followed. Verification was followed by a discussion with the reconciliation team. An optical layout check was also necessary because the translation often had an impact on the layout. This was then followed by testing of the scoring mechanism (cf. chapter 2.2.3) and finally by testing the integrated assessment tests.

For this multi-step localisation process, extensive documentation of all changes and comments is indispensable, as also highlighted by the

[5] XLIFF is the abbreviation for XML Localisation Interchange File Format. It is a standard file format which permits making adaptable data editable and manageable within a localisation process (Savourel et al. 2008).

International Test Commission Test Adaptation Guidelines6 (Hambleton and de Jong 2003, p. 130):

> "I.1 When a test is adapted for use in another population, documentation of the changes should be provided, along with evidence of the equivalence."

In PIAAC, so-called Verification Follow Up Forms (VFF)7 were used, which contained the afore-mentioned translation and adaptation guidelines and provided space for discussion for the different people involved in the translation process. The verifier who checked the reconciled version could add comments and recommendations to one or several parts of the translation, and the country's reconciliation team could respond by accepting or refusing the verifier's recommendations. In the VFFs, the different players could also explain the reasons and motives for their decisions. Thus, for each country detailed documentation was generated that contained a summary of the decisions made for every single localisation issue. Errors or changes that were valid for all countries were compiled in a special "errata sheet" available for all countries.

In practice, translators (or reconcilers or verifiers) were only able to translate the text derived from the test items and made available for them in the aforementioned XLIFF file. Everybody involved in the translation process could preview the English source version of the test item on a web-based Item Management Portal. More importantly, they were also able to interact with the item in the way the test participant would during the test (e.g. they could answer the item, click on links within the stimulus, see all different pages that were included in items that simulated webpages etc.). After translating an XLIFF file or after correcting a translated version, it was possible to upload the translation to the portal and preview the translation there. For the translation of the XLIFF text, the Open Language Tool (OLT)8 was used. The OLT includes a Translation Memory, which helps to maintain consistency across test units.

2.2.2 Localising the layout

Layout adaptations became necessary after translating despite all efforts made during the internationalisation process described in section 2.1. Country teams were required to check all their items for potentially corrupt layout and report these issues to the consortium, which then tried to adapt the layout as required by the country. This resulted in a protracted exchange of communication between all partners involved until all problems were taken care of.

As mentioned in section 2.1.2, the source version provided extra space to accommodate languages whose translations take up more space than English does. In several cases, the allocated space was still insufficient and had to be extended (or resulted in a smaller font). For languages that took up less space than the source version, the layout had to be adapted in a few cases as well.

In a few isolated cases, graphics had to be exchanged in a localised item for authenticity reasons (for example, an image that shows bottles had to be exchanged when the beverage itself was not known in the country or carried specific connotations).

Also, justified text - which looked like, for example, an authentic newspaper article in the source item - looked unusual in some translations because the languages had much longer word lengths than the English original. This problem was solved by hyphenating words. In such cases, hyphenated text was not included in text that was crucial for answering the item. All of these issues (and more) were discussed and checked by item experts to ensure that they would not compromise cross-country comparability.

2.2.3 Localising the scoring

In a computer-based test, a respondent can provide answers in several ways: response types can include multiple choice, short text entry, numeric entry, selection of radio buttons or combo boxes, text field entry, highlighting text, marking graphical objects or cells, and many more. The entries given by the respondent then have to be scored. Scoring items means that a score is assigned to the test participant's response. The score is defined by a scoring rule, which relates (ranges of) responses to scores. Automatic, machine-based scoring requires defining scoring rules within the system. Manual scoring, by human experts, relies on scoring guidelines including scoring rules and assignments of typical responses to scores.

[6] These guidelines were set up to support test item developers when adapting test intstruments (Hambleton and de Jong 2003).
[7] cApStAn, a linguistic quality control company, was responsible for the generation of VFFs and for the general translation and adaptation procedures. Their verifiers were responsible for checking the translated versions produced by the country teams.
[8] The OLT is an open-source tool that is available online (The Source for Java Technology Collaboration n.d.).

Most response types, with the exception of free text entry, can be automatically scored by a computer system in a straightforward manner. Automatic scoring can be more efficient than human scoring as the time-consuming work by human scorers is not necessary. Whenever adaptive testing9 is used, automatic scoring becomes a pre-requisite.

In a test that has to be translated, adaptation of the scoring usually does not pose any difficulties for response types such as multiple-choice or marking graphical objects or cells. Here it is most important that the text is translated. The choice of correct or incorrect responses usually does not change their location (in most languages) and no further adaptation is necessary. However, some scoring information is language or country specific and has to be prepared in a way that allows for its localisation.

One example is the scoring of numeric responses, for instance in the case of items involving currencies. An item might ask the respondent to calculate the price of a purchase, e.g. "This radio costs 30 dollars. How much does it cost when a 10 % discount is given?". The correct response is "27 dollars" in our example. If the price of the radio and the correct response are not adapted in a country with a different currency (for example, Japan where 1 USD = approx. 80 Yen), the item context is no longer authentic. In PIAAC (in contrast to PISA where the fictional currency zeds is used, as mentioned earlier), real currencies were retained, with guidelines for adaptation. In such a case, the localisation of the scoring content becomes inevitable and the defined correct response will have to be changed in the system.

Localising scoring rules of numeric entry items requires not only the definition of the correct number(s) but also decisions about acceptable spelling formats for numbers (e.g. with respect to the kind of decimal separator). Although there are international standards for number formats defining the spelling of numbers country by country, it may be too strict to accept only responses as correct if they adhere to these standards. Given considerable variability in the usage of number formats within countries (and even within test participants), a more lenient scoring approach that accepts alternative number formats was judged to be more appropriate for PIAAC.

In PIAAC, complexities also arose from the adaptation and localisation of the highlight response mode. For the highlight response mode, the respondent has to mark the correct answer in the stimulus text to indicate his or her answer. Here is an example to illustrate this and to explain how the scoring mechanism is designed in PIAAC:

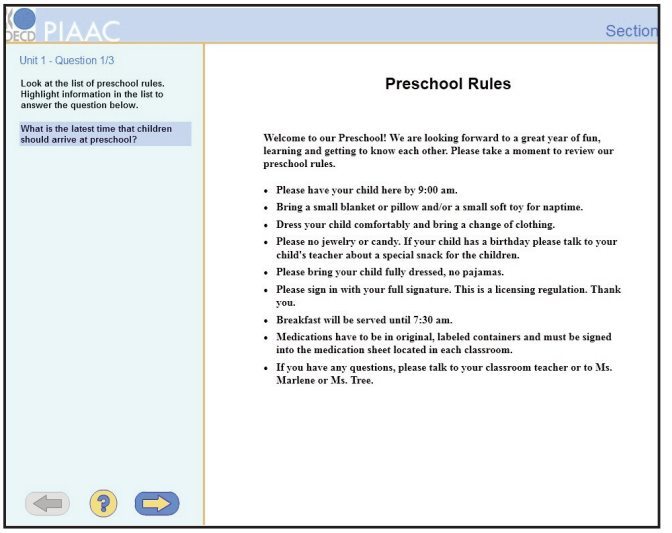

Figure 2: Preview of a sample highlight item

[9] Adaptive testing means that the number pattern of correct and incorrect responses of a respondent has an impact on the difficulty of the next test items that are presented. The idea is that a test taker that repeatedly shows low level skills is more likely to receive easy items, while a respondent that shows high level skills is more likely to receive difficult items. So in Computerized Adaptive Testing the item difficulty is tailored to the individual's performance level. Too hard and too easy items which would not contribute to a reliable measure are avoided. In an adaptive test, the upcoming item or set of iems is selected adaptively based on the performance shown in previous items. In some instances, e.g. for selecting the first item (set), additional contextual information (e.g. educational level) may be used as well (Wainer 2000).

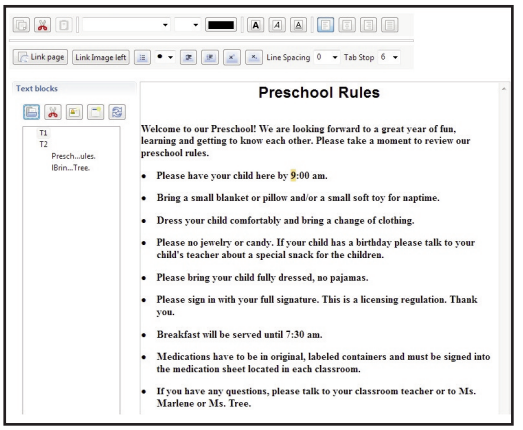

Figure 3: The interface of the CBA Itembuilder. The correct text block T1 is highlighted.

The respondent is given a text and he is asked to highlight information in the text to give his answer. The question refers to the stimulus text and asks: "What is the latest time that children should arrive at preschool?" (cf. Figure 2).

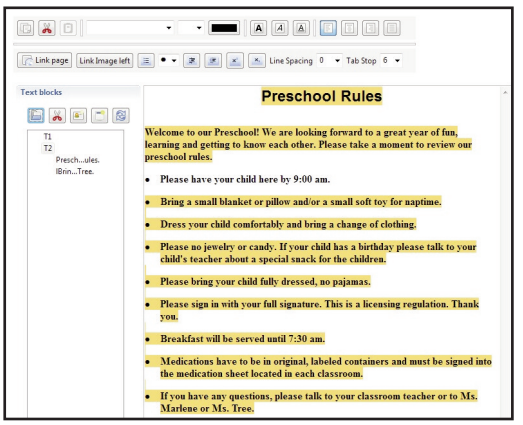

Figure 4: The interface of the CBA Itembuilder. The miss area text block T2 is highlighted

The respondent gives the correct answer by highlighting the number "9". To "teach" the computer system which answer is correct and which is incorrect, the item developer has to indicate in the stimulus itself what the correct and what the incorrect answer is. This is done by defining text blocks and by specifying scoring rules referring to these text blocks.

The number "9" becomes a part of the "minimum correct response" text block.

In our example, the item developer makes the number a part of T1.

The remainder of the text becomes text block T2 (as shown in figure 4). Note that the sentence in which the correct answer is included is left out of any of the text blocks.

The following scoring rules are defined in the authoring tool:

Hit = complete (T1)
Miss = partial (T2)

This means that the answer given by the respondent is considered to be correct when

1. The whole of T1 has been selected.
2. No part of T2 has been selected.

Text that is not included in any of the text blocks CAN be selected. It is a part of the so-called maximum correct response (which is "Please have your child here by 9:00 am.").

After the translation of the text content, it is important that the text blocks are redefined as well, because they are language-dependent and thus unlikely to match the source version in terms of size and location. In PIAAC, this followed reconciliation and subsequent check by the verifiers. For the adaptation of the text blocks the "Translation Textblock Editor" was used, a standalone tool derived from the CBA ItemBuilder mentioned above.

Countries could not define new text blocks or delete text blocks, but they were able to adapt the content of the text blocks according to their needs. This process required several informed decisions about how to localise the scoring rule in a comparable way as illustrated with a simple example.

Question: What does the text say about how much computer scientists earn?

Stimulus text:

> "Computer scientists under 30 typically make more than the average salary for their age from day one."

In the source version of the item, the minimum correct response text block consists of "more", "than" and "average". Once the text is translated into German, "average salary" becomes "Durchschnittsgehalt". Should test participants receive a correct response when they only highlight "Durchschnitts" (which represents "average" in this

compound noun)? Scoring experts within the countries had to find answers to many scoring-specific questions, e.g. how to deal with compound words; how to deal with endings (e.g. should inflections be included in the minimum correct response?); is the correct response still comparable to the source version when the target version includes significantly more words in its minimum correct response?

After the field test, the text blocks could be re-adapted if the field test results showed that items in one country behaved differently from items in other countries. The localisation of scoring was a difficult task for the countries.

3. Lessons learned and open questions

This paper, so far, has given a brief introduction to LSA projects and discussed the role of localisation in the area of LSA studies. As previously described, localising tests for international LSA studies poses specific challenges that are not necessarily encountered in other localisation processes. One of the main differences concerns the struggle between authenticity and comparability when localising, and the adaptation of scoring information. By describing a real scenario, we examined how these aspects are dealt with in practice. PIAAC is special in its own right because it is the first international LSA study that is completely computer-based (with a paper-based option for inexperienced computer users). The multi-step procedure that was implemented to manage these difficulties poses some open challenges for future studies. Many of these challenges result from the shift to a computer-based test mode and can be classified into two categories: firstly, new difficulties concerning the localised content and, secondly, and more importantly, difficulties regarding the internationalisation and localisation process when trying to master both complexity and quality assurance. These challenges will be described in the following paragraphs.

With regard to the test content, special linguistic difficulties arise within the new field of test items that simulate technology-rich environments (web pages, software tools,…). The question of authenticity arises when web content is translated into languages with a low population of speakers, like Estonian: a stimulus mimicking a web page might be considered as inauthentic if completely translated into languages for which only limited content is available on the web. Also, people in some countries typically do not use their national language as an application interface language (for example because the localised interface was only introduced very late and people were already used to working with an English interface). Hence, the question arose as to whether the interface language should be translated or not. Similar concerns can arise for languages with different fonts (for which it is difficult to translate URLs in a web browser). Not translating this content might make the item more difficult for respondents who are less familiar with using a computer (or do not speak English). Translation, however, might make the item inauthentic, which might have an impact on the difficulty of the item as the technical terminology might be less familiar to the test taker. Similar problems can arise when tests are translated into minority languages (like Valencian or Basque). Even though inauthenticity might be less of a problem for speakers of these languages (as many of them are familiar with using their language in new contexts), there might also be an impact on the difficulty of the tasks. These issues and their influence on an item's validity of measurement will have to be discussed further in the future.

The shift from a paper-based to a computer-based test mode has a significant impact on the adaptation processes. One big difference compared to the adaptation process for paper-based tests is the separation of adaptable content from static, non-adaptable content. On the one hand, this makes the process more complicated and requires many case-by-case decisions. On the other hand, it automatically brings many issues to light that would not necessarily be (knowingly) identified during a localisation process for paper-based tests (Should the inline formatting be exactly the same across languages? Can the font size be changed? What degree of freedom is allowed for changing layout?). In addition, the computer-based mode of test items technically facilitates the direct comparisons of localised test items. Hence, the shift presents a challenge as well as an opportunity for making localisation issues more visible than before.

This also leads to the problem of finding the right balance between flexibility and control. In PIAAC, a conscious decision was made not to allow the countries or the software to make any changes to the layout. As previously mentioned, this was helpful because the consortium (and the item developers) maintained control over the location of the text. On the other hand, it is questionable whether it would not

have been preferable to allow for more decentralised layout adaptations. If the size of a text box automatically adapted to the length of the translated text, many of the manual adaptations of the items (bearing the risk of introducing new errors) could have been avoided. Especially for languages like Korean and Japanese, it would also be helpful if countries were granted more flexibility to adapt some selected elements of the layout manually. Line space, for example, had to be doubled for Korean because the Korean characters become illegible with the default line space set in the source items. For Japanese, line breaks were also an issue: there are no blanks between characters and text is usually justified. When designing the translation process and the software tools for the translation process, these requirements should play a role from the very beginning and be a part of the items' internationalisation process. Certain countries would thus gain access only to selected layout elements that could not be dealt with during internationalisation.

The adaptation process for computer-based tests also requires the ability to integrate two additional steps into the localisation process, i.e. layout and scoring adaptations. Defining the sequence of the adaptation steps becomes a challenging task in such a complex process. For example, allowing any linguistic changes to be made after the completion of scoring and layout adaptations means that these adaptations have to be re-checked. An ideal approach would be to first complete all linguistic changes, and secondly resolve all layout issues. The scoring should be adapted at the very end. Since the localisation of automatic scoring rules is a new area in LSA, and the consequences of scoring adaptations are not visible in the item itself, countries need to test scoring carefully following a test plan.

It also became clear that it is important that all people involved in the adaptation process are able to interact with the item in the same way as the test participant. This also became apparent for the scoring mechanism, for the adaptation of which it was crucial to be able to test all changes by trying to give correct and incorrect responses. Countries received detailed test cases from the consortium giving the correct or incorrect responses for the source version, which could then be adapted by the country and checked on the Item Management Portal by giving the required response. The portal then gave feedback on whether the response was correct or incorrect. This allowed for immediate feedback on whether the adaptations (of e.g. text blocks) resulted in the desirable scoring behaviour. This procedure - testing while adapting - made the scoring adaptation process efficient for countries because they received immediate feedback for any scoring adaptation decision.

Another challenge regarding the efficiency of the localisation process refers to the question of who should make adaptations, i.e. whether certain adaptation steps should be centralised and done by experts in the consortium, or de-centralised and become the responsibility of the national teams. For instance, at the beginning of the project, the consortium tried to give countries the freedom of adapting their numeric scoring. This decision was made because the people in the national teams would be able to decide if items that include currencies should be adapted or not (cf. previous section). However, it soon became clear that it was not efficient to teach this complicated adaptation procedure to all countries: input was needed from numeracy experts to decide whether changing a currency number would change the item's psychometric properties, such as difficulty, as well as from technical experts to implement the changed scoring rule. In PIAAC, this process was then modified and centrally organised: the consortium and the numeracy expert group made recommendations and gave feedback regarding certain problematic items, the country groups made sure that the items were authentic for their country, and the consortium made the technical implementation. A conclusion from the PIAAC case study is that it is more efficient to implement technically difficult adaptations centrally, after countries have provided input as regards authenticity.

One step to allow for more de-centralisation and transparency regarding content decisions would be to give countries more, and broader, information as a basis for making decisions and finding solutions during the localisation process. For instance, as a future enhancement of the PIAAC approach, one might try to make information regarding localisation issues available and useable across countries, so that each country team can gain a new cross-country perspective and is able to compare different localisation problems and solutions. The bundling of information could result in a more consistent approach and increased quality. Many localisation problems do not only exist for one language but across languages. In these cases it would be very helpful for a country's translation and localisation team if they had an overview of all the localisation problems that emerged for an item in other countries.

Furthermore, they could check whether they might have a similar problem that they are not yet aware of. In addition, once a problem is identified, they could directly check solutions other countries had found for a similar problem and use these solutions as a guideline for their own decision. A technical solution for such a centrally available cross-country information and documentation pool would be needed for the localisation process.

Source version management is a difficult issue in an adaptation process that includes many different partners in many different countries. Even though the source items, after "internationalisation', are supposed to be final prior to starting the localisation process, several issues are only found once countries have started on their translations, and more are found through the verification procedure. One problem regarding the file-based solution in PIAAC was that every time a new version became available, countries had to download this version and check that this was the latest version. A lot of these issues can probably be avoided by advance translation: This is done in PISA, for example, where two source versions are created: a French source version is developed in parallel with the English source version. At least some of the issues that concern the translatability of items can thus be identified in advance. There are fewer errors when the source versions are released for translation by the countries (Grisay 2003). Still, it is likely that not all problems can be found, even by using advance translation. Source version management itself could be technically supported by using a content management system, which would prevent subsequent errors caused by miscommunication between partners or overlooking changed material.

The question of source version management leads to the question of translation version management. The multi-step localisation procedure also made it difficult for countries to translate because they had to consult and edit a lot of material for translation. This should be reduced so that cycling between many documents is not necessary anymore; a technical solution should be found. A first step in this direction has been made with the framework of PISA 2012 computer-based testing, whereby item-specific translation/adaptation guidelines and comments by the different players (translators, reconciler, verifier, country post-verification reviewer) are carried within the XLIFF file rather than being presented in a separate monitoring form.

4. Conclusion

As described in this article, many problems have to be tackled in LSA studies that are not usually present in localisation processes where comparability does not play a role. In particular, localisation in LSA studies deals with balancing between authenticity in each country and comparability across countries.
To handle this challenge, a multi-stage translation and verification approach is pursued, including:

- Preparing internationalised test material
- Localising content (text, images)
- Localising layout
- Localising meta-data, e.g. scoring rules.

Still, several aspects can be transferred to other localisation processes as well. For instance, the issue of version management is of general importance, as well as the question of when to test a localised version. Other domains for which the quality of translations is highly critical might also benefit from the multi-stage translation and verification process that is used for LSA studies. Similarly, the question as to which adaptations should be done, and by whom, is also relevant in all localisation processes.

On the other hand, LSA studies can take more advantage of the advances made by the localisation industry. As LSA studies are shifting from paper-based assessment to computer-based assessment, the time seems right to move towards commonly used standards and tools. In PIAAC, the first steps in this direction have been taken by introducing the XLIFF standard as a basis for the translation and by requiring countries to use a translation memory (TM) aware translation tool such as the OLT. Nevertheless, not all of the new possibilities have been tried yet. Another promising approach is to put more emphasis on source content quality assurance.

References

Freedle, R. (1997) The relevance of multiple-choice reading test data in studying expository passage comprehension: The saga of a 15 year effort towards an experimental/correlational merger, Discourse Processes, 23(3), 399-440.

Grisay, A. (2003) Translation procedures in OECD / PISA 2000 international assessment, Language Testing, 20(2), 225-240.

Hambleton, R. (2002) 'Adapting Achievement Tests into

Multiple Languages for International Assessments', in Porter, A. and Gamoran, A (eds)., Methodological advances in crossnational surveys of educational achievement, Washington, DC: National Academy Press, 58-79.

Hambleton, R., de Jong, J. (2003) Advances in translating and adapting educational and psychological tests, Language Testing, 20(2), 127-134.

Hambleton, R., Merenda, P. and Spielberger, C., eds. (2005) Adapting Educational and Psychological Tests for Cross-Cultural Assessment, Mahwah, NJ: Erlbaum.

Kirsch, I. (2001) The International Adult Literacy Survey (IALS): Understanding what was measured, ETS Research Report RR-01-25, Princeton, NJ: Educational Testing Service.

LISA (2003). Localisation Industry Primer, 2nd ed., Fechy, Switzerland: The Localisation Industry Standards Association (LISA).

McQueen, J., Mendelovits, J. (2003) PISA reading: cultural equivalence in a cross-cultural study, Language Testing, 20(2), 208-224.

Mullis, I., Martin, M., Ruddock, G., O'Sullivan, C., Preuschoff, C. (2009) TIMSS 2011 Assessment Frameworks, Amsterdam: International Association for the Evaluation of Educational Achievement (IEA).

OECD (2010) PISA: Programme for International Student Assessment, OECD Education Statistics (database).

OECD (2011) PISA 2009 Results: Students on Line: Digital Technologies and Performance, Volume VI, PISA, OECD Publishing.

Savourel, Y., Reid, J., Jewtushenko, T.; Raya, R.M. (2008) XLIFF Version 1.2 [online], available: http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html [accessed: 24 June 2011].

Schäler, R. (2007) Reverse Localisation, Localisation Focus, 6(1), 39-48 .

Guidelines for Authors

Localisation Focus
The International Journal of Localisation
Deadline for submissions for VOL 11 Issue 1 is 30 June 2012

Localisation Focus -The International Journal of Localisation provides a forum for localisation professionals and researchers to discuss and present their localisation-related work, covering all aspects of this multi-disciplinary field, including software engineering and HCI, tools and technology development, cultural aspects, translation studies, human language technologies (including machine and machine assisted translation), project management, workflow and process automation, education and training, and details of new developments in the localisation industry.

Proposed contributions are peer-reviewed thereby ensuring a high standard of published material.

If you wish to submit an article to Localisation Focus-The international Journal of Localisation, please adhere to these guidelines:

- Citations and references should conform to the University of Limerick guide to the Harvard Referencing Style
- Articles should have a meaningful title
- Articles should have an abstract. The abstract should be a minimum of 120 words and be autonomous and self-explanatory, not requiring reference to the paper itself
- Articles should include keywords listed after the abstract
- Articles should be written in U.K. English. If English is not your native language, it is advisable to have your text checked by a native English speaker before submitting it
- Articles should be submitted in .doc or .rtf format, .pdf format is not acceptable
- Article text requires minimal formatting as all content will be formatted later using DTP software
- Headings should be clearly indicated and numbered as follows: 1. Heading 1 text, 2. Heading 2 text etc.
- Subheadings should be numbered using the decimal system (no more than three levels) as follows:

  Heading
  1.1 Subheading (first level)
  1.1.1 Subheading (second level)
  1.1.1.1 Subheading (third level)
- Images/graphics should be submitted in separate files (at least 300dpi) and not embedded in the text document
- All images/graphics (including tables) should be annotated with a fully descriptive caption
- Captions should be numbered in the sequence they are intended to appear in the article e.g. Figure 1, Figure 2, etc. or Table 1, Table 2, etc.

More detailed guidelines are available on request by emailing LRC@ul.ie or visiting www.localisation.ie