Localisation Workflows
THE IMPACT OF PROCESS WELL-HANDEDNESS ON AUTOMATION

CS6033 DISSERTATION
UNIVERSITY OF LIMERICK
MASTER OF SCIENCE MULTILINGUAL COMPUTING AND LOCALIZATION

SUPERVISOR: DR. DAVID FILIP

Nicolas Martinez
Submitted to the University of Limerick, November 2013
Declaration

I hereby declare that this is entirely my own work and that it has not been submitted for the award of any degree at any other university.

DocuSigned by:

Nicolas Martinez

06/11/2013
Abstract

Localisation Workflows: The impact of process well-handledness on automation

Author: Nicolas Martinez

In the context of an ever growing need for LSP to optimize their offerings to their clients, TMS stands out as a seemingly obvious choice to deliver higher quality content faster and cheaper.

As the trend is to deploy TMS workflows full speed ahead, do those systems cater well to different use-cases as they can be found in traditional localisation processes, involving frequent updates, change of scope at language or file level as localisation is tied in more closely than ever to content authoring?

We started with the assumption that TMS only supported well-handled use-cases with fixed sets of well-defined transitions and limited support to non-well-handled cases with flexible unpredictable transitions. As a corollary, we deemed that perhaps certain use cases are not suitable for automation in TMS.

Thus, the objective set forth with this paper was to evaluate the strengths and limitations of TMS using the concept of workflow well-handledness.

To this avail, we began by looking at the current state of affairs in traditional localisation workflows and found that, although the traditional translation, editing and proofreading cycles and stakeholders follow a defined model, there is a need for added flexibility at each step to optimise turnaround times.

We then positioned TMS in the supply chain, looked closely at localisation business processes and in what ways solutions based on TMS could improve current localisation workflows. It became apparent that TMS is not a monolithic system and workflow engineers need to closely work with project stakeholders and be aware of all process applications to design, implement and ensure relevant reporting.

The practical steps to deploy a workflow in TMS around key process areas and specific goals were detailed, revealing that there should be just enough human tasks to cater for change in requirements and the system must be robust enough to allow time-critical maintenance on running production workflows.

Workflow patterns were introduced allowing us to scrutinise this flow around complementary perspectives, that of control-flow, data, resource and exception handling.

Finally, we put the concept of well-handledness to the test in a case-study of industrial Lionbridge TMS using workflow patterns. The case study confirmed that in control-flow patterns support is rather basic and limited to well-handled use-cases, with a fixed set of well-defined transitions and predictable resources. This is mainly because such systems focus on the end-user to meet SLAs rather than offering powerful underlying workflow.

A key take away was that support for non-well-handled use-cases is unlikely to change in the near future. Rather, improved workflow patterns support may be partially provided, but it likely will be based on well-handled workarounds.
Table of Contents

Acronyms Used.................................................................................................................. v
Glossary ............................................................................................................................... vii
List of Figures..................................................................................................................... ix
List of Tables....................................................................................................................... ix
Introduction......................................................................................................................... 1
Limitations of this study ..................................................................................................... 3
Chapter 1 : Traditional Localisation processes ............................................................... 4
Chapter 2 : The role of TMS in Business Processes .......................................................... 9
  2.1 The Zachman framework ontology ............................................................................. 11
  2.2 The Enterprise Content Management (ECM) methodology ................................. 14
     2.2.1 The content lifecycle ...................................................................................... 16
        2.2.1.1 Acquisition ............................................................................................... 17
        2.2.1.2 Storage ................................................................................................... 17
        2.2.1.3 Delivery ................................................................................................. 18
     2.2.2 Valuing content lifecycle.................................................................................. 19
     2.2.3 Content Maturity Model.................................................................................... 19
        2.2.3.1 People ...................................................................................................... 20
        2.2.3.2 Process .................................................................................................... 20
        2.2.3.3 Systems .................................................................................................. 21
     2.2.4 Architecture and technology............................................................................. 21
     2.3 Definition of localisation automation black box and its boundaries................. 23
        2.3.1 Business intelligence and reporting.............................................................. 24
        2.3.2 TMS performance analysis ......................................................................... 24
     2.4 Business Process Automation .............................................................................. 26
     2.5 Process improvement using CMMI .................................................................... 27
Chapter 3 : Process for upgrading a workflow to TMS .................................................. 30
  3.1 Requirements development....................................................................................... 31
3.1.1 [SG 1] Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements

3.1.2 [SG 2] Customer requirements are refined and elaborated to develop product and product component requirements

3.1.3 [SG 3] The requirements are analysed and validated

3.2 Integrated Project Management

3.2.1 [SG 1] The project is conducted using a defined process tailored from the organization’s set of standard processes

3.2.2 [SG 2] Coordination and collaboration between the project and relevant stakeholders are conducted

3.3 Measurement and analysis

3.3.1 [SG 1] Measurement objectives and activities are aligned with identified information needs and objectives

3.3.2 [SG 2] Measurement results, which address identified information needs and objectives, are provided

3.4 Project Monitoring and control

3.4.1 [SG 1] Actual project performance and progress are monitored against the project plan

3.4.2 [SG 2] Corrective actions are managed to closure when the project’s performance or results deviate significantly from the plan

3.5 Technical solution

3.5.1 [SG 1] Product or product component solutions are selected from alternative solutions

3.5.2 [SG 2] Product or product component designs are developed

3.5.3 [SG 3] Product components, and associated support documentation, are implemented from their designs

Chapter 4: Well-handledness in localisation workflows

4.1 Petri nets

4.1.1 Workflows
4.1.2 Net completion ................................................................. 45
4.1.3 Net well-handledness ........................................................... 46
4.2 Well handledness in TMS ....................................................... 46
4.3 Workflow patterns ............................................................... 50

Chapter 5: Case study of Lionbridge TMS .................................... 52
5.1 Background ........................................................................ 52
5.2 Terms and basic definitions .................................................. 56
5.3 Limitations ........................................................................ 61
  5.3.1 Abstract syntax limitations .............................................. 61
    5.3.1.1 No Vertical Modularization ..................................... 61
  5.3.2 Control-Flow limitations .................................................. 62
    5.3.2.1 Limited support for human decision points ................ 62
    5.3.2.2 No run-time control to skip steps in workflow ............. 63
    5.3.2.3 No custom branching ............................................. 65
  5.3.3 Resource limitations ........................................................ 66
    5.3.3.1 No run-time loop with historic based allocation .......... 66
  5.3.4 Data limitations .............................................................. 67
    5.3.4.1 Limitation in post work log generation ..................... 67
    5.3.4.2 No Correlated files support for Human activities .......... 67
    5.3.4.3 No source/target pair data integrity checks ............... 68
  5.3.5 Exception handling limitations ......................................... 69
    5.3.5.1 Basic support for task errors handling .................... 69
    5.3.5.2 No explicit timeout for external applications ............. 71
    5.3.5.3 No runtime support for external triggers ................. 71
  5.4 Evaluation using workflow patterns ..................................... 59

Chapter 6: Conclusion .............................................................. 73
Bibliography ........................................................................... 76
Appendices ............................................................................. 80
List of appendices

Appendix A  Typical localisation process using BPMN .........................................................80
Appendix B  Lionbridge approval to use TMS for this paper .............................................82
Appendix C  State of the art Lionbridge process using Petri nets .................................83
Appendix D  Toolbox in Lionbridge TMS Workflow Designer ........................................86
Appendix E  TMS progression example (Lionbridge 2013) ............................................87
Appendix F  Pivot process in TMS (Lionbridge 2013) .....................................................88
Appendix G  Control flow patterns TMS evaluation .........................................................89
Appendix H  Exception handling patterns TMS evaluation .............................................95
Appendix I  New project Engineering template ..............................................................97
<table>
<thead>
<tr>
<th>Acronyms Used</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BPA</td>
<td>Business Process Automation</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Management</td>
</tr>
<tr>
<td>BPMN</td>
<td>Business Process Model and Notation</td>
</tr>
<tr>
<td>CMM</td>
<td>Capability Maturity Model</td>
</tr>
<tr>
<td>CMMI</td>
<td>Capability Maturity Model Integration</td>
</tr>
<tr>
<td>CMS</td>
<td>Content Management System</td>
</tr>
<tr>
<td>DTP</td>
<td>Desktop publishing</td>
</tr>
<tr>
<td>ECM</td>
<td>Enterprise Content Management</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GG</td>
<td>Generic Goal</td>
</tr>
<tr>
<td>GMS</td>
<td>Globalisation Management System</td>
</tr>
<tr>
<td>GP</td>
<td>Generic Practise</td>
</tr>
<tr>
<td>HB</td>
<td>Hand Back</td>
</tr>
<tr>
<td>HO</td>
<td>Hand Off</td>
</tr>
<tr>
<td>IFS</td>
<td>Integrated File System</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator</td>
</tr>
<tr>
<td>LFS</td>
<td>Lock Frequent Segments</td>
</tr>
<tr>
<td>LSP</td>
<td>Language Service Provider</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>MLV</td>
<td>Multilanguage Vendor</td>
</tr>
<tr>
<td>MT</td>
<td>Machine Translation</td>
</tr>
<tr>
<td>PA</td>
<td>Process Area</td>
</tr>
<tr>
<td>PM</td>
<td>Project Manager</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>RSS</td>
<td>Rich Site Summary</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SG</td>
<td>Specific Goal</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SP</td>
<td>Specific Practice</td>
</tr>
<tr>
<td>TAT</td>
<td>Turnaround Time</td>
</tr>
<tr>
<td>TEP</td>
<td>Translation Editing and Proofreading</td>
</tr>
<tr>
<td>TM</td>
<td>Translation Memory</td>
</tr>
<tr>
<td>TMS</td>
<td>Translation Management System</td>
</tr>
</tbody>
</table>
# Glossary

The glossary has been collated from various sources with terms such as “Cloud computing”, “Localisation”, “Process”, “SaaS” and “Workflow” taken from Cameron (2011), the term “Black box” taken from (Daniel et al. 2011), the terms “Bitext” or “Token” taken from Filip & Conchúir (2011) and finally the terms “Process area” were taken from CMMI Institute (2013). The remainder are my own definitions.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bitext</strong></td>
<td>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 &quot;unclean&quot; 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.</td>
</tr>
<tr>
<td><strong>Black box</strong></td>
<td>In software testing, approach of study where the inner architecture of programs is not examined.</td>
</tr>
<tr>
<td><strong>BPMN</strong></td>
<td>Graphical representation of business processes in a business process model.</td>
</tr>
<tr>
<td><strong>Bug</strong></td>
<td>A software or hardware defect</td>
</tr>
<tr>
<td><strong>Cloud computing</strong></td>
<td>Cloud Computing is made possible by the establishment of virtual private networks (VPNs) that can be used and accessed by the organisation to serve its customers.</td>
</tr>
<tr>
<td><strong>Localisation</strong></td>
<td>The process of adapting a software products or services for different languages, countries, or cultures. In addition to language considerations, such as support for foreign character sets, localisation may require adaptations for currencies, time zones, national holidays, cultural assumptions and sensitivities, dialects, colour schemes, and general design conventions.</td>
</tr>
<tr>
<td><strong>Localisation kit</strong></td>
<td>Set of instructions with reference materials (optional) to enable the localisation process on a provided set of files.</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Petri net</strong></td>
<td>Mathematical model used as a process modelling technique</td>
</tr>
<tr>
<td><strong>Pre-processing</strong></td>
<td>This consists of taking the source files, marking up the files using a set of rules to separate translatable content from non-translatable and generating XLIFF based files to send to translators.</td>
</tr>
<tr>
<td><strong>Post-processing</strong></td>
<td>In a usual localisation cycle, after translation, the files are generally post-processed before being delivered to the client.</td>
</tr>
<tr>
<td><strong>Process</strong></td>
<td>Procedure consisting of logically connected steps with predefined inputs and outputs.</td>
</tr>
<tr>
<td><strong>Process area</strong></td>
<td>CMMI models define a “process area” as a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making improvement in that area.</td>
</tr>
<tr>
<td><strong>SaaS</strong></td>
<td>Software as a Service, where the users connect to the application over the internet and do not need to install or setup anything on their own machine.</td>
</tr>
<tr>
<td><strong>Token</strong></td>
<td>Whatever travels through a defined process or a 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.</td>
</tr>
<tr>
<td><strong>Workflow</strong></td>
<td>Automation of business processes, in whole or in part, where documents, information or tasks are passed from one participant to another for action, according to a set of rules. A business process is a logically related set of workflows, work steps, and tasks that provides a product or service to customers.</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1: The Waterfall Model (Zaki 2013) ................................................................. 4
Figure 1.2: Example of a Typical Localisation Workflow (Peris 2013) ......................... 6
Figure 1.3: Localisation testing life cycle (EVERS 2012) ........................................... 8
Figure 2.1: The Automated Localization Workflow Reference Model (i18N Inc. 2005) ................................................................................................................................. 9
Figure 2.2: The Zachman Framework for enterprise architecture (Zachman International, Inc. 2012) ........................................................................................................... 11
Figure 2.3: The project lifecycle (Cameron 2011) .......................................................... 15
Figure 2.4: The content lifecycle (Cameron 2011) .......................................................... 16
Figure 2.5: Content maturity model (Cameron 2011) .................................................... 19
Figure 2.6: Schema of a black box (Wikipedia 2012) .................................................... 23
Figure 4.1: The chunking and reassembling activities in a typical localisation process. (Filip & Conchúir 2011) ........................................................................................................ 48
Figure 5.1: Internal TMS Template Implementation (Lionbridge 2013) ......................... 52
Figure 5.2: Blank Lionbridge TMS Workflow Designer workflow template .............. 58
Figure 5.3: Standard workflow in Lionbridge TMS Workflow Designer ..................... 61
Figure 5.4: TMS job task options ......................................................................................... 62
Figure 5.5: Flow control options in activity Properties using TMS Workflow Designer ................................................................................................................................. 62
Figure 5.6: Custom parallel branching concept ............................................................... 65
Figure 5.7: Sequential activities in TMS .......................................................................... 65
Figure 5.8: Failed activity alert in TMS .......................................................................... 69
Figure 5.9: Alert control options in TMS ........................................................................ 70

List of Tables

Table 2.1: LSP application responsibility matrix .............................................................. 22
Table 2.2: Process Areas (Entinex, Inc. 2013) relevant to TMS deployment ............... 28
Table 2.3: Capability level 1 GG and GP (Entinex, Inc. 2013) ................................... 29
Introduction

Nowadays, Language Service Providers (LSP) have a growing requirement for automation in their localisation processes and they are looking towards solutions like Translation Management System (TMS) or Machine Translation (MT) to reduce operating costs, increase quality and reduce turnaround times.

The flexibility of TMS to cater to multiple use-cases, as they occur in traditional localisation processes, can be measured by the workflow modelling concept of well-handledness.

In practise, a large class of processes can only be modelled with nets that properly complete, but are not well-handled. This situation can arise as soon as the constraints on the order of process steps are anything other than a choice between fixed paths


Previous research, such as workflow patterns conducted by the Workflow Patterns Initiative (2012) or Modeling Business Processes by Aalst & Stahl (2011) focused on workflow management systems, not TMS in particular. Also, well-handledness, although often implied in the workflow patterns definitions, was not directly a subject of study.

Current research led by Adam Morera Mesa at the Localisation Research Centre (LRC) on Workflow recommendation in Localisation seems pertinent to the topic of this paper, as he discusses tasks involved in localisation workflow and workflow patterns, but his work is out of scope\(^1\). This being said, the optic of this paper is different as it seeks to identify the shortcomings of current TMS offerings when presented with various localisation workflow use-cases.

\(^1\) Due to logistic and time constraints and the fact that this work was not published at the time the literature review was conducted, it is entirely out of scope of this paper.
Henceforth, this thesis will attempt to critique TMS around the concept of workflow well-handledness. Does TMS cater well to non-well-handled use cases?

In an attempt to start a discussion aimed at answering this question, we shall begin our research with the following assumptions:

• Processes need to be well-handled in TMS as they require a fixed set of well-defined transitions,

• TMS may support non-well-handledness in localisation processes to some extent, and

• Certain use cases are not suitable for automation in TMS.

We will address the validity of each point above in this paper.

We begin our research, in Chapter 1, by defining the concept of traditional localisation processes, looking at typical workflows and defining stakeholders.

Chapter 2 will highlight where TMS lies in the supply chain. To do so, we will look closely at localisation business processes and in what ways solutions based on TMS can improve current localisation workflows.

Assuming that a TMS has now been deployed in the LSP, the next logical step in Chapter 3 is to define the necessary steps to build a localisation workflow around TMS, using the process improvement methodology defined in the previous chapter.

The next step in Chapter 4 consists in introducing the mathematical concept of workflow well-handledness at an abstract level and looking at its application to TMS. Additionally, we will define the workflow patterns methodology, distinguishing various perspectives to assess the performance of a TMS.

Finally, Chapter 5 will consist of an evaluation of the Lionbridge TMS using workflow patterns focusing on the concept of process well-handledness, both defined in Chapter 4. This survey will use the process areas and specific goals defined in Chapter 3 as a guide.
Limitations of this study

This paper will draw conclusions based on the examination of Translation Management Systems (TMS) in traditional localisation processes from the exclusive point of view of Language Service Providers (LSP). This approach is meant to study typical localisation workflow use cases in today’s localisation industry. New use cases such as those introduced by crowdsourcing will not be studied.

As a corollary of this study focusing solely on workflow management systems, translation memory management systems, such as SDL TM Server solution, are out of scope.

---

2 Due to limited time, only Lionbridge TMS will be examined. SDL TMS / SDL WorldServer and GlobalSight won’t be discussed in this paper.
Chapter 1: Traditional Localisation processes

Traditional localisation processes may involve human translation or machine translation. All use-cases are generally well defined and widely used in today’s localisation factories. Social or community localisation involve very different use-cases and are not part of the scope of this paper.

By traditional localisation process, I refer to processes that involve Translation, editing and proofreading (TEP) and generally follow a Waterfall Model where each stage must be completed before the next one occurs. This is inspired from software design where:

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In waterfall model phases do not overlap.

(Kumar 2013)

Looking at the Waterfall Model (Figure 1.1), localisation is just one part of software development.

Figure 1.1: The Waterfall Model (Zaki 2013)
In the localisation world where translation was traditionally requested towards the end of software development (Figure 1.1), where the product is finished and most localisable content is final, a waterfall model as described by Kumar (2013) often proved adequate and sufficient to handle most localisation needs. Naturally, the majority of today’s TMS workflows are designed based on such traditional localisation processes.

Of course, it must be noted, although this paper will not discuss Agile localisation, that:

More and more software companies are adopting Agile development methodologies in an effort to have truly global product launches that ship simultaneously in multiple languages. Consequently, decision makers at localization companies are being forced to adapt their traditional business models.

(Zaki 2013)
Figure 1.2: Example of a Typical Localisation Workflow (Peris 2013)
Figure 1.2 represents a traditional LSP translation workflow which brings together several human and machine elements:

- Project Manager (PM) for quote management, project scheduling and query management
- Translation vendors or freelancers
- Engineering team for pre and post-processing of files
- Review or proofreading teams
- Desktop publishing (DTP) team if non-text localisable assets needs to be extracted for translation.

Appendix A introduces a more basic model than the one introduced in Figure 1.2, without DTP involved, but outlining who does what using BPMN.

The minimum requirement for a decent TMS must be to support such basic localisation workflows.

After translation is complete, it is typical to have at least one round of localisation testing conducted by the LSP, the client or a third-party. During testing, the translated text will be integrated in the product and tested in-context by team of language testers. Any changes (or bugs) will be either implemented there and then (bug fixing) or a list is sent to the LSP for fixing in the localised files and integration into their final TM. The combination of translation and testing workflows can be seen in Figure 1.3 below.
Figure 1.3: Localisation testing life cycle (EVERS 2012)
Chapter 2: The role of TMS in Business Processes

It is essential to define where TMS lies in the supply chain. TMS are localisation workflow management systems responsible for handling process automation.

In theory, the workflow engine is a general purpose tool that allows arbitrarily complex processes to be defined: any number of steps, any number of people involved anywhere in the world, any task be it manual or automated. In practice, there is a trade-off between complexity and ease-of-use: some systems have a fixed sequence of steps and the only configuration possible is to skip a step; other systems support general workflow templates with conditionals and have their own scripting language.

(i18N Inc. 2005)

Figure 2.1: The Automated Localization Workflow Reference Model (i18N Inc. 2005)
Let us consider the graphic in Figure 2.1 above which describes a typical localisation workflow. The cycle starts at the source level from a Content Management System (CMS) where the content is created, for instance at the customer's side, and ends with target localised files saved onto the same or another CMS.

In-between this cycle, at the LSP premises, there is a Globalisation Management Systems (GMS) which uses one or more inter-connected modules to manage the localisation process such as:

- External customer-facing translation job submission portal
- TMS to manage the internal localisation workflow automation
- TM, glossaries to centralise linguistic assets
- Separate translation engines such as machine translation (MT)
- PM billing module

In such an automated workflow, typically, the TMS is the backbone of the translation effort. It is invoked by the change detection module when new files are detected. Subsequent steps from Job creation to Job Archival are encoded in the TMS underlying workflow template design and settings, connecting to additional modules as needed such as retrieval or updating the TM, performing MT.

Now that we have illustrated where TMS lies in the supply chain, let’s see where it all fits in the global company strategy.
2.1 The Zachman framework ontology

The Zachman framework (Figure 2.2) is an ontology, or in other words a specification of a conceptualization, for describing a business. Using a matrix diagram, we will use it to define the structure of LSPs by asking basic questions (columns) to look at their components from different perspectives (rows), that of business ideas, technology operation and end-users.

The software and hardware infrastructure of an LSP are typically service-oriented as made of several modules. As we showed earlier, GMS are composed of several interconnected modules, not a monolithic single-tiered architecture.

Figure 2.2: The Zachman Framework for enterprise architecture (Zachman International, Inc. 2012)
LSPs typically offer multiple services such as translation and testing to businesses interested in deploying a product or service in another region of the world to that their content has been authored for.

The Zachman framework in Figure 2.2, adapted to a LSP could be classified as below:

- **What**: The translation assets inventory.
- **How**: Identify and analyse localisation processes.
- **Where**: Where are the assets stored and how do they get distributed in and out of the company firewall.
- **Who**: Define and understand each responsibility node for all client accounts including human and system elements.
- **When**: Look at the agreed turnaround times with LSP clients as per Service Level Agreement (SLA) and how efficient are the underlying workflows to this goal.
- **Why**: The motivation element; look at what value does the company currently offer in relation to the competition and benefits of current technology.

Now that we have defined some core classifications behind an LSP, in order to get a bird's eye view on the current state of affairs in the company, each one needs to be looked at from different perspectives corresponding to different model names:

- **Scope Contexts / Executive Perspective**: Using high level analytics from the business operations for planning the company strategy.
- **Business Concepts/Business Management Perspective**: The operation managers have detailed technical view on how the business is run day to day in practical terms and how concepts translate in terms of technology.
- **Architect Perspective/System Logic**: Solutions architects are the interface between the client and the engineer/technician. Technically savvy, they are assigned to one or more clients and gather requirements from each client so as to design a customised solution.
- **Technology Physics/Engineer Perspective**: Workflow engineers may be tasked to design a workflow and set up a process for the day to day
engineer to execute and build the workflow based on the architect’s recommendations. They have in-depth technical knowledge of localisation processes.

- **Tools Components/Technician Perspective:** Tools developers or support, generally not part of the localisation staff but on the IT side. They work closely with the engineer to troubleshoot or develop custom solutions.

- **Operations Instances/Enterprise Perspective:** All day-to-day ground staff working on the localisation production chain which comprises generally of:
  - PM for scheduling and quoting
  - Language coordinators supporting translators in answering linguistic queries and preparing localisation kits
  - Localisation engineers executing any file-based pre and post-processing tasks
  - DTP
  - QA testers who test the translated products in-context.

The above should, by no means, be an exhaustive list of criteria applicable to all LSPs as several roles may well be merged in smaller sized companies. This should serve our purpose in conceptualising the structure of a typical medium to large size LSP.

In chapter section 2.2, we will draw from the Enterprise Content Management (ECM) methodology to outline transformative trends in LSPs.
2.2 The Enterprise Content Management (ECM) methodology

We have seen the Zachman Framework which helped us define the state of affairs in an LSP. Enterprise Content Management (ECM) is an organisational process methodology. It is complementary to the Zachman Framework in the following way:

The Framework (ontology) is a STRUCTURE whereas a methodology is a PROCESS. A Structure is NOT a Process. A Structure establishes definition whereas a Process provides Transformation.

(Zachman International, Inc. 2011)

We can use this methodology to determine where TMS fits in this transformation. This will help us later, in chapter section 2.3, define the localisation automation black box and its boundaries.

ECM techniques will help us for the following reasons:

ECM practice catalogues the many mechanisms for information capture, management and distribution. It attributes cost and assesses the value of retention. It advocates collaborative processes to reduce process bottlenecks and capacity issues. ECM strategy addresses the differing throughputs of the organisation to ensure that there is no information overload. It defines the roles which an organisation needs to ensure its knowledge and the information it is based on remain consistent and up-to-date.

(Cameron 2011)
Figure 2.3 illustrates the high-level ECM stages to consider when developing a new strategy such as introducing a TMS.

Figure 2.3: The project lifecycle (Cameron 2011)
2.2.1 The content lifecycle

Let us consider the content lifecycle which can be broken down into the three areas, Acquisition, Storage and Delivery which can be looked at from three different factors as you can see in Figure 2.4 below:

Cameron (2011) considered the content object as the lowest denominator. In the localisation industry focused on TMS, the content types of importance would be but not limited to:

- Linguistic assets such as translation units in a TM
- Localisable or reference files
- Process documents
- Metrics logs.

To rephrase words quoted by Cameron (2011) and adapt it to the localisation industry, the link between acquisition and delivery is a mark of success when developing localised content. TMS allow to optimise this link through automation.
2.2.1.1 Acquisition

- Enterprise acquisition: various acquisition methods. Among the most popular, File Transfer Protocol (FTP), email or cloud computing portals.
- Content acquisition: new projects from client comprising of source files (text files or graphic files), reference materials and instructions. Graphic files will need to be transformed by the DTP team.
- Acquisition management: Localisation requests hand-off mechanism and schedule is agreed by the PM with the client; at least initially for each new type of content to set up an enterprise acquisition process.

2.2.1.2 Storage

- Enterprise storage: Once acquired, client hand-off is generally either stored on Integrated File System (IFS) if acquired by email or FTP. Alternatively, hand-off files can be automatically sent to TMS repository if acquired by cloud computing portals. As a bridge between IFS and TMS, it is also possible to have a custom batch solution to perform automated tasks (i.e. pre-processing) when detecting new content at a given IFS path.
- Content storage: Graphic files (which get sent to DTP for text extraction) and text files (which need to get pre-processed automatically or by a localisation engineer) generally follow a different path resulting in different storage rules.
- Storage management: The IT department generally sets permission and retention rules for content stored on IFS. Version control on IFS can be managed with specialised tools. In TMS, such parameters are accessible at the workflow level or inherited at the system level.
2.2.1.3 Delivery

- Enterprise delivery: Localised content hand-back is often delivered the same way it has been acquired, either by email, FTP or by cloud computing portals to the client CMS.

- Content delivery: The client expects localised files to follow a given template, generally closely based on the source format. For instance if graphic files have been sent for localisation, it is unlikely they will expect text files back but instead the corresponding localised graphics, so DTP would be the final step to integrate localised text back in the graphic container.

- Delivery management: Like for the hand-off, the PM will generally perform the delivery back to the client or at least work to arrange a suitable delivery mechanism. The client would determine if deliveries have to be staggered (i.e. deliver each language after it is completed) or made after localisation tasks are completed for all languages. All the above would be encoded in the TMS workflow.

The information stated in the above nine areas only serves to outline a typical content solution including TMS for a typical LSP.
2.2.2 Valuing content lifecycle

Now that we have defined the content lifecycle for medium to large LSPs, we need to determine a way to sustain and measure benefits.

Organisations which understand the relationship between their content objects can start to determine which content objects bind, and attract stakeholders to, their repository. By doing so, they enhance the stickiness of their content overall. Key performance indicators (KPIs) should be created to enable a measure to be included in corporate reports.

(Cameron 2011)

Any TMS should be able to generate KPI reports based on production metrics. Such KPIs can be created based on data points such as:

- Number of localisation jobs in production per client
- Job word count
- Job languages
- Reuse of content from the TM
- Times (PM, engineer, translation, etc.)

2.2.3 Content Maturity Model

Each step in accomplishing the ECM goals requires an assessment of where the organisation is. In line with business case planning there must be some means to measure improvements necessary to sustain the programme.

(Cameron 2011)

The content maturity model will allow us to perform this assessment. It examines all parts of the business from three important components: people, processes and systems.

Figure 2.5: Content maturity model (Cameron 2011)
Most medium to large size LSPs interested in a TMS have reached the Enterprise stage (Figure 2.5). There is little optimisation in the company but they consider a TMS as a way to remedy this situation with processes becoming candidates for optimisation. They are aware of the value of their content and thus know what is required to create value.

2.2.3.1 People
The core staff (engineers and project managers at minimum) are conscious of the need to document processes, reuse information and classify it. This is to ensure consistency in the work quality and reducing turnaround times as well as costs to remain competitive on the localisation market. Members of staff work together between different departments to design or improve localisation processes.

2.2.3.2 Process
Processes are clear and consistent with clear responsibilities for change management. Processes can be analysed to identify room for improvement. Workflows are designed toward customer satisfaction.

The organisation is using enterprise resource planning (ERP) systems at certain levels although they are rarely connected directly to GMS where the localisation production happens. Localisation process statistics are gathered generally directly from one or more of the GMS modules (such as the TMS or the TM).

Corporate IT supports intranet (i.e. Microsoft SharePoint) and the IFS to give a standard repository for all exchange.

Processes also outline clear levels of responsibility for all localisation activities.
2.2.3.3 Systems

The benefit of SaaS is understood at system level. Information ubiquity is part of the company's global strategy to interface the two sides of the company firewall with technology such as extranet bug databases, TMS, online translation portals or content delivery platforms.

Systems are designed or configured to be adaptable to different levels of network or systems load to support the company growth. The support of those systems is split between dedicated teams of software engineers and Corporate IT with a trend towards dedicated teams listening closely to the end-users of the system for perpetual improving and system stability.

2.2.4 Architecture and technology

Creating viable software architecture requires the agreement of all stakeholders from both the business and technical communities. It demands a vision which can be shared and communicated at many different levels and from many different perspectives. Ultimately an architect must be a leader who can call on many allies within the organisation: a technical and considered sage, a decisive and persuasive advocate of the architecture.

(Cameron 2011)

In an LSP, the first phase for a solutions architect, tasked to determine the suitability of a TMS to replace a traditional process, is preparation. This consists of the following tasks:

- Review and selection of audit focus areas, tasks and required outputs, based on an Initial analysis, interviews and understanding of the specific challenges and goals the customer faces.

(Moravia 2013)

The solutions architect must gather requirements from each stakeholder. It is not uncommon for the lead localisation engineer, who have a more detailed technical perspective, to drive the requirements gathering. However, there are other stakeholders who need to be appropriately consulted such as the PM.

The following matrix details each stakeholders need so as to allow the solutions architect gather requirements from the relevant stakeholders for each need.
Table 2.1 is an example of such responsibility matrix outlining all applications involved in the localisation effort at the operational level. They need to be considered together when deploying a new solution such as a TMS. It is based on the structure of the Dublin division of the LSP Lionbridge Technologies, the best performing LSP in 2013 according to Common Sense Advisory (DePalma & Hegde 2013) but this structure can be generalised to many LSPs.

Table 2.1: LSP application responsibility matrix

<table>
<thead>
<tr>
<th>Application suite</th>
<th>Department / Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation workflow</td>
<td>Localisation Engineer</td>
<td>Managing file preparation rules / File QA process.</td>
</tr>
<tr>
<td>Translation workflow</td>
<td>PM</td>
<td>Project scheduling / quoting.</td>
</tr>
<tr>
<td>TM / Terminology database</td>
<td>Language Lead</td>
<td>Managing translation assets quality.</td>
</tr>
<tr>
<td>ERP</td>
<td>Project Coordinator</td>
<td>MLV invoice tracking / Billing management.</td>
</tr>
<tr>
<td>MT</td>
<td>MT Engineer</td>
<td>MT engine customisation.</td>
</tr>
<tr>
<td>Translation productivity platform</td>
<td>Localisation Engineer</td>
<td>Pre-processing and post-processing text files.</td>
</tr>
<tr>
<td>Translation Management Portal</td>
<td>PM</td>
<td>Coordinating hand-off and hand-back with third-party clients.</td>
</tr>
<tr>
<td>Translation tools</td>
<td>Localisation Engineer</td>
<td>Ensuring files are translation-ready.</td>
</tr>
<tr>
<td></td>
<td>Translator / Reviewer</td>
<td>Translating or correcting files / QA before delivery as instructed by language coordinator.</td>
</tr>
<tr>
<td></td>
<td>Language coordinator</td>
<td>Reviewing the files when drafting localisation kit or when translation is done.</td>
</tr>
<tr>
<td>Query database</td>
<td>Language coordinator</td>
<td>Supporting MLV queries.</td>
</tr>
<tr>
<td>Bug database</td>
<td>Localisation Test Engineer</td>
<td>Tracking of localised software bugs.</td>
</tr>
<tr>
<td>Desktop publishing software</td>
<td>DTP</td>
<td>Extracting text / Creating localised assets.</td>
</tr>
</tbody>
</table>
Table 2.1 takes into consideration the applications and departments involved in all operational activities. Those match all stages of the GMS as shown in Figure 2.1.

However, the Zachman framework applied to the LSP structure in chapter section 2.1 outlined that the Executive Perspective and Business Management Perspective should also be taken into account in the state of affairs. Therefore, it is also important to consider the application requirements from higher organisational levels so as to provide relevant business intelligence and reporting to those stakeholders.

### 2.3 Definition of localisation automation black box and its boundaries

Let us first clarify what is the concept of a black box (Figure 2.6) and why this approach is of interest here:

The black box concept here is derived from the black box method of software testing, where the inner architecture of programs is not examined. [...] We propose a black box approach for the performance analysis, as this allows us to study the systems as they are perceived by the actual users of the system, the perspective we are particularly interested in this work.

(Daniel et al. 2011)

![Figure 2.6: Schema of a black box (Wikipedia 2012)](image)

The localisation automation black box is the TMS itself, one of the modules of the GMS illustrated in Figure 2.1 and the focus of the paper.

We have defined the stakeholders in chapter section 2.2.4 with Table 2.1 in particular looking inside the black box with the users of the systems and the application domain.

In doing so, we have defined the characteristics of the black box. It needs to be well defined and validated by those stakeholders during deployment or upgrading to avoid causing ripple effects outside of its boundaries. If such adverse effects are observed, the black box needs to be redefined.
2.3.1 Business intelligence and reporting

Business intelligence and reporting data is important for upper management as we have mentioned in the end of chapter section 2.2.4. The process of deploying a TMS might break some valuable data entry points. This is why automation requirements need to be carefully gathered.

For our purpose, we have to differentiate between global high-level business metrics for managers, accessed within ERP systems from outside of the black box and the projects metrics used by project stakeholders such as the PM to regulate the flow of day to day operations.

The solutions architect designing the migration path has to work closely with relevant management stakeholders or ERP systems architects outside of the black box to develop customised automation to ensure that the data flow is in the expected format. This would include:

1. Export reporting data from TMS
2. Design custom automation to manipulate the data into a format accepted by the ERP
3. Submit the data to the ERP

2.3.2 TMS performance analysis

To justify the deployment or upgrade of a TMS, one has to look at KPIs of translation workflows.

Let us define the notion of workflow from a technical point of view and its proposed evaluation through the concept of a black box:

A workflow is the automation of a business process where atomic work units (task) are assigned to participants (agent) according to a workflow schema (process model). A workflow management system (WfMS) manages several process instances (cases), and relies on a database management system (DBMS).

We propose an approach to evaluate the performances of a WfMS treating it as a black box and a monolithic system purely observed from outside.

(Daniel et al. 2011)
"KPIs characterize the performance of the measured system and considers several layers of one service. Typically, the most relevant KPIs at the IT infrastructure level consider CPU, CPU idle time for I/O, CPU load (length of the process), main memory usage, disk throughput, and network bandwidth" (Daniel et al. 2011)

In parallel, the performance of a TMS should be evaluated using the same KPIs that would be used when valuing content lifecycle in traditional localisation workflows as listed in chapter section 0 such as:

- Throughput in terms of translation jobs and words processed
- “Completion Time. This indicator is the average amount of time required to complete a case, measured as the interval between the start time and the completion time of process execution.” (Daniel et al. 2011)

Using those KPIs, it should be possible to compare TMS performance with that current performance of localisation workflows and measure the added value of such systems.
2.4 Business Process Automation

The underlying reason for deploying a TMS is to reduce turnaround times and cost whilst increasing quality by the means of business process automation.

With direct translation costs being largely fixed, attention often focuses on reducing non-translation costs.

(Moravia 2013)

However, it must be noted that “Enterprise-level translation management systems cater well for their well-defined use cases.” (Filip & Conchúir 2011).

In their paper, Filip & Conchúir (2011) have concluded that all localisation processes have been found to possess the following lowest common denominator: “Parsing of source text -> routing Bitext -> enriching Bitext -> quality assuring Bitext -> exporting target text”

This content transformation rule is at the core of TMS and ought to be helpful to translate traditional localisation process activities to their TMS workflow equivalent.

Whilst doing so, it must also be noted that, “as business process automation is considered, evaluations should be made, not only of where automation can contribute but also those business processes where it is critical to the enterprise’s business model to have human touch-points and decision-making.” (Shacklett 2011). Indeed, not all activities in a localisation workflow can or should be automated and best judgement should be made consulting with the relevant stakeholders.

Finally, “the BPA approach states that until a process is automated, there is no real value in analysing and defining it, and that the cycle of business change is so rapid there simply isn’t time to define every process before choosing which ones to address with automation, and that delivering immediate benefits creates more value.” (Wikipedia 2013). It is not hard to imagine when you consider that many of the stakeholders have to provide their input or implement automation whilst still working on their primary day to day production tasks.
2.5 Process improvement using CMMI

Looking at process quality is paramount when deploying a new software solution:

Today it is globally accepted that the quality of the process used to develop a specific product has a strong impact on its final quality. Therefore, the improvement of development processes is not just a possible option but a survival strategy.

(Vates S.A. 2011)

Process improvements methodologies such as the Lean Six sigma methodology or Capability Maturity Model Integration (CMMI) by Carnegie Mellon University can align well with a Business Process Management (BPM) view, "as they constantly look for incremental opportunities to make processes more efficient and reduce defects." (Wikipedia 2013)

CMMI is designed to seek improvement in definite process areas. “Six Sigma, on the other hand aims at solving specific product or process related issues within the context of overall organizational process improvement.” (Nayab 2011)

Also, Six sigma paid reports are esoteric in nature whereas CMMI reports are public. Thus, CMMI seems rather a better candidate for this paper to evaluate the localisation process in the context of a TMS deployment. Indeed TMS domain is more restricted as opposed to GMS or ERP made of multiple modules encapsulating a wider part of the organisation.

Let us consider our existing scenario of an LSP looking to deploy TMS. The black box defined in chapter section 2.3 represents the process areas to evaluate using CMMI.

One may categorise relevant process areas (PAs) for TMS deployment as per Table 2.2, applicable to our black box defined in chapter section 2.3.
### Table 2.2: Process Areas (Entinex, Inc. 2013) relevant to TMS deployment

<table>
<thead>
<tr>
<th>Process Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements</strong></td>
<td>The purpose of requirements Development (RD) is to elicit, analyse, and establish customer, product, and product component requirements.</td>
</tr>
<tr>
<td><strong>Integrated</strong></td>
<td>The purpose of Integrated Project Management (IPM) is to establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.</td>
</tr>
<tr>
<td><strong>Measurement &amp; Analysis</strong></td>
<td>The purpose of Measurement and Analysis (MA) is to develop and sustain a measurement capability that is used to support management information needs.</td>
</tr>
<tr>
<td><strong>Project</strong></td>
<td>The purpose of Project Monitoring and Control (PMC) is to provide an understanding of the project’s progress so that appropriate corrective actions can be taken when the project’s performance deviates significantly from the plan.</td>
</tr>
<tr>
<td><strong>Technical Solution</strong></td>
<td>The purpose of Technical Solution (TS) is to design, develop, and implement solutions to requirements. Solutions, designs, and implementations encompass products, product components, and product-related lifecycle processes either singly or in combination as appropriate.</td>
</tr>
</tbody>
</table>

We are interested in the Continuous representation of the CMMI model which “allows organisations to pick any number of process areas, and also pick to whatever depth of capability they want to become in those process areas.” (CMMI Institute 2013)
In the context of this paper, all we are interested in is the improvement factor and not appraisals. Therefore, we will not discuss how the model is organised. Instead, in order to improve the LSPs operational performances as part of a TMS workflow deployment, we will examine in Chapter 3 each Process Area (PA) against the Generic Goal 1 in Table 2.3.

Generic Goals (GGs) can be defined by one or more Generic Practices (GPs): Every Process Area (PA) has at least one Specific Goal (SG), made up of at least two Specific Practices (SPs). The SPs in any PA are unique to that PA (Entinex, Inc. 2013)

Each GG corresponds to a Capability Level defined during the appraisal process to determine the compliance of a particular PA.

Table 2.3: Capability level 1 GG and GP (Entinex, Inc. 2013)

<table>
<thead>
<tr>
<th>Generic Goal 1 [GG1]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generic Practice 1.1 [GP 1.1]:</strong></td>
<td>Perform the specific practices of the process to develop work products and provide services to achieve the specific goals of the process area.</td>
</tr>
</tbody>
</table>

The process supports and enables achievement of the specific goals of the process area by transforming identifiable input work products to produce identifiable output work products.
Chapter 3: Process for upgrading a workflow to TMS

There are many aspects to consider when considering managing a workflow in TMS. The trend in LSPs is clearly towards bringing in automation at every turn so it is important to consider the limitations of the current workflows and of the TMS itself.

It is best practice to evaluate what other companies have done before deciding to upgrade your workflows. GlobalSight Eight Steps to a Successful TMS Roll-Out below provide a good high-level breakdown:

- Step 1: Find the pain
- Step 2: Audit your workflows
- Step 3: Define what can be automated
- Step 4: Define what needs to be managed outside the TMS
- Step 5: Relieve the biggest pain first
- Step 6: Test before deployment
- Step 7: Pace the roll-out
- Step 8: Address Issues – fast

(Rock 2010)

We will now use the CMMI methodology, not to evaluate, but to propose a plan of action for bringing a standard workflow, such as defined in Chapter 1, into TMS.

We will do so using the information gathered by applying other ontologies and methodologies defined in the Chapter 2, specifically:

<table>
<thead>
<tr>
<th>The Zachman framework applied to an LSP</th>
<th>Chapter section 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>The content lifecycle of an LSP</td>
<td>Chapter section 2.2.1</td>
</tr>
<tr>
<td>TMS KPIs defined in Valuing content lifecycle and performance analysis</td>
<td>Chapter section 0 and chapter section 2.3.2</td>
</tr>
<tr>
<td>Enterprise level Content Maturity Models centred around people, process and systems</td>
<td>Chapter section 2.2.3</td>
</tr>
<tr>
<td>LSP application responsibility matrix</td>
<td>Table 2.1</td>
</tr>
</tbody>
</table>

Having defined the scope of the black box in chapter section 2.3, we will attempt to define each identified Process Area (PA) in Table 2.2 against the associated Specific Goal (SG) using CMMI for Development.
3.1 Requirements development

3.1.1 [SG 1] Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements

As stated in chapter section 2.1, workflow engineers may be tasked to design a workflow and set up a process for the day to day engineer to execute. They have in-depth technical knowledge of localisation processes.

Rock (2010) argues that finding pain spots, showing the teams in what way TMS can alleviate that pain and a roll-out in small increments to reduce the adjustment factor is the key to a successful adoption of the system.

Therefore, the focus needs to be on the future TMS users from the very start. The workflow engineer needs to make sure that each user is on-board, that perception issues are managed correctly whilst planning to adapt the current workflow to be compatible with TMS. Those teams and their tasks have been defined in the LSP application responsibility matrix in chapter section 2.2.4, Table 2.1.

In parallel, the workflow engineer needs to audit the existing workflow and again the key is, as often, to “think and work in modules or incremental steps” (Rock 2010) or in other words to break the problem down whilst thinking about the solution as a whole. Indeed deploying a TMS is often not just about the workflow itself but also about any required connectors, say to push hand-off (HO) files to TMS.

Whilst analysing the workflow documentation, which might be out of date, and any documentation sent by the main requestor, generally the PM or the lead engineer, it will be necessary to consult with the teams to find out exactly who does what in the workflow to establish and prioritise the customer requirements.
TMS optimises the link between acquisition and delivery through automation. We previously defined the content lifecycle of an LSP in section 2.2.1. This should be valuable to define customer requirements:

- The *Enterprise* factor gives us information about the current state of affairs on how the content is transformed and where to, during acquisition (HO), storage and delivery (HB). This should be read in parallel with the *Systems* component of the content maturity model in chapter section 2.2.3
- The *Content* factor defines the nature of the files, what is acquired, stored and delivered. This is required to build file filters during the Technical solution PA in chapter section 3.5. This should be read in parallel with the *Process* component of the content maturity model in chapter section 2.2.3

### 3.1.2 [SG 2] Customer requirements are refined and elaborated to develop product and product component requirements

After having defined customer requirements, the workflow engineer needs to translate those into requirements for the necessary TMS components or modules. As mentioned earlier, TMS may just be part of the solution and the customer requirements may apply to TMS connectors as well (i.e. HO and HB tools).

We mentioned in chapter section 2.4 that “Enterprise-level translation management systems cater well for their well-defined use cases.” (Filip & Conchúir 2011). Also, process use-cases generally need to be well-handled to be good candidates for automation. We will discuss this concept in detail in Chapter 4.

Therefore, the workflow engineer needs to carefully map customer requirements to each module and define how each module interface with each other.

During this PA goal, optimisation goes hand in hand and it is necessary to define what can be automated as well as what needs to be managed outside of TMS, simplifying the workflow, merging existing process steps or removing any of those steps that don’t add any value or are redundant. (Rock 2010)
File transfers are processes that often are good candidates for process automation. So are many routine localization project management tasks, such as the identification of project files, the assembly of project instructions or the transmission of files to and from vendors.

(Rock 2010)

In the WorldServer case-study by LSP VistaTEC, they stated that TMS can only be deployed on select projects:

While the 'Idiom team' familiarized themselves with the nuances of the platform, we set out to identify a viable pilot project. We defined the following criteria:

- Must have either (a) consistent repeatable deliveries, or, (b) large volume split across a number of files.
- Continuous or numerous updates throughout.
- File types and their internal formats need to stay the same.
- Preference for XML, html and java property files.
- Well designed and established process, i.e. one that cannot change mid-translation.

Many steps can be automated with WorldServer, including the following:

- Extraction of translatable text from files
- Scope analysis
- Preparation of localization kits
- Handoffs between steps
- Updating of Translation Memory (TM)
- Updating of multilingual terminology glossaries
- Generation of target language files.

(VistaTEC 2013)
3.1.3 [SG 3] The requirements are analysed and validated

Those requirements should be used to model a few concepts and scenarios

Requirements need to be analysed to balance the TMS users’ needs and the constraints of the system, validating the proposed workflow concept against the production environment and making sure requirements are sufficient and necessary. (CMMI Institute 2010)

Especially if considering replacing human steps by automated activities, the workflow engineer must be careful how to design such steps and validate this with the actual human who is executing this step in the current workflow.

3.2 Integrated Project Management

3.2.1 [SG 1] The project is conducted using a defined process tailored from the organisation’s set of standard processes

Figure 1.2 represents a traditional LSP translation workflow. The PM would usually base any requests for improvement or new projects setup on such standard workflow. Moreover, it is common for medium to large LSPs to have frameworks to assure quality and on-time deliverables to customers, such as Lionbridge LEO. Such a framework gives access to templates to define and measure processes, as detailed in the People component of the content maturity model of chapter section 2.2.3.

Work environments and content exchange platforms are usually standardized and managed by Corporate IT. Such practices are encapsulated by the Management factor of the content lifecycle in project section 2.2.1 which defines who is involved, when and their role on content acquisition, storage and delivery.

An essential element that affects the definition of a TMS project plan is the Service Level Agreement (SLA). The When dimension of the Zachman framework defined in chapter section 2.1 covers this aspect.

Teams should be established from relevant departments pertaining to the project. All day-to-day ground staff working on the localisation production chain have been defined in chapter section 2.1 in the Operations Instances/Enterprise Perspective level of the Zachman framework.
Successful TMS deployment, TMS scripts or related automation connectors should be shared back to the organisational process assets pool so that it can be re-used by other workflow engineers and thus reduce subsequent implementations as well as ensure consistency within the enterprise.

3.2.2 [SG 2] Coordination and collaboration between the project and relevant stakeholders are conducted

To manage the involvement of the teams established in the previous goal, the project manager should refer to the LSP application responsibility matrix established Table 2.1 as part of chapter section 2.2.4

During TMS implementation, the PM should work closely with the workflow engineer to identify, negotiate, and track critical dependencies as well as resolve issues with relevant stakeholders. (CMMI Institute 2010)

3.3 Measurement and analysis

3.3.1 [SG 1] Measurement objectives and activities are aligned with identified information needs and objectives.

When valuing the content lifecycle of traditional localisation workflows in chapter section 2.3.2, we defined the KPIs to evaluate the TMS performance. If comparing against the same metrics applied to the current localisation workflow before automation in the TMS, it is possible to measure the added value therein.

Measurement data are obtained and stored (CMMI Institute 2010) using reporting tools integrated in TMS:

There should be a choice of download formats, and a way to create custom reports based on queries to the underlying database. Client-side users don’t always need visibility into the linguistic side of the system. What they do need is flexible on-demand reporting. Whether there is a live dashboard or a facility to run reports, the system has to be customisable so the reports respond to their Organization’s needs both from the point of view of Program Management and Financial tracking.

(Peris 2013)
This measurement data may then be communicated to a reporting module in the CMS or eCommerce platform using content connectors that leverage this data using the TMS API. This could be used in conjunction with additional automation giving “the possibility to monitor FTP locations and file systems.” (Peris 2013)

The task of the workflow engineer or TMS expert is generally limited to configuring the TMS settings to ensure the workflow generates the measurement data and in the expected format. Specialised staff such as CMS experts or tools developers are then involved during any the connector configuration or programming phase.

3.3.2 [SG 2] Measurement results, which address identified information needs and objectives, are provided.

Once measurement data are accessible to the Executive or Business Management perspectives described in our application of the Zachman Framework to the LSP model in chapter section 2.1, they may be need for reconfiguration of the reporting data. This especially would be required if the TMS workflow is reconfigured due to a change in the business strategy such as for example replacing human translation by MT with post-editing. In that case, it will be required to measure the impact of this change in terms of turnaround time, quality and cost.

Measurement data, measurement specifications, and analysis results will need to be managed and stored (CMMI Institute 2010), but this is out of scope of this paper.

3.4 Project Monitoring and control

3.4.1 [SG 1] Actual project performance and progress are monitored against the project plan.

Project plan parameters have been documented when valuing the content lifecycle of traditional localisation workflows in chapter section 0. Their values can generally be monitored on the TMS in the form of global project statistics such as job average turnaround time (TAT), average effort word counts, percentage of on time deliveries and percentage of system tasks as opposed to human tasks.

Within an actual project, activity logs determine how to bill the client and how to pay your vendors on per project or per file basis.
On the TMS dashboard, it should be possible to track project status in real-time. There, the PM can see the active jobs, tasks not assigned or not accepted and tasks in work, at any given time. This contrasts with the normal manual project planning and tracking using Microsoft Project in traditional workflows not using the TMS. Indeed, in those workflows, project statistics are generally entered based on stakeholders’ feedback, usually communicated by email in the course of the project.

In case action is needed if a translator un-claims files or if any other errors occur during the workflow execution, the TMS generally generates alerts visible on the project dashboard. Additionally, the system could be setup to generate an email or the PM may access those by connecting to RSS feeds to get instant notifications.

Generating quotes for translators is generally managed automatically by the TMS which is pre-configured on standard quoting practise from the organisation:

> Workflow systems are used by Enterprises which have a regular stream of translation requirements. Vendor rates are usually pre-agreed rather than negotiated on a project by project basis. The Workflow technology should be able to hold these rates and automatically generate a quote when a translation request is submitted. It should be able to handle minimum charge, PM fees, volume discounts and provide a workflow step where the requestor or authorised user can accept or reject the quote.

(Peris 2013)

The main risk when involving humans in a workflow is reliability. Managing schedule is a core component of a TMS. You can work with your stakeholders to determine their availability during your project dates. This schedule can be entered in the TMS and the workflow can manage user assignments based on that.

In cases where MT is required for a project, it should be possible to use aforementioned defined metrics to evaluate the impact of using MT in the workflow. This is required for the Measurement and analysis PA. Such impact can be measured, for example, in terms of post-editing cost and other human stakeholders’ time involvement.
After translation is complete, billing is generally accomplished outside of TMS. However, let us consider a scenario where multiple translators work on the same projects and save their work in a common TM and thus benefit from each other's work. The most accurate compensation for this work should not be based purely on the quote which was generated before starting the work but instead on TM leverage statistics. Maybe you want to pay a lower amount for any 100% exact matches, fuzzy matches (less than 100%) that would have been added in the TM from the time the quote was issued by other translators working on the same project? Of course this approach should be agreed with the translators or MLVs. The point is that to bill in such way would require the TMS to leverage this statistics from the Bitext translated file, if XLIFF based.

3.4.2 [SG 2] Corrective actions are managed to closure when the project’s performance or results deviate significantly from the plan.

Just like traditional manual localisation workflows, the system must be robust enough to allow time-critical maintenance on running production workflows.

Finally, escalation is an important feature i.e. if a task gets stuck, will it be re-routed to someone in charge, and will an alert be sent to whoever is in charge? If not, then the system must be monitored very regularly to make sure everything is on track.

(i18N Inc. 2005)

Any issues identified during workflow execution, and not just the ones where a workaround could be found, should be notified to the engineer in charge of maintaining the workflow in operation. Such issues may then be escalated to the workflow engineer who originally designed the TMS workflow if needed. The idea is to fix any problems as they occur and not to wait until they re-occur.
Such problems could include, but are not limited to, server errors (i.e. time out) when trying to contact external applications for processing of certain activities in the workflow. Indeed, TMS is often configured to communicate directly with external applications such as TM / Terminology database, MT engine, Translation productivity platform or Translation Management Portal. For details on these applications, please refer to Table 2.1.

In case of issues with external applications, TMS should generate alerts with details of such as the error message and/or the application debug log.

### 3.5 Technical solution

#### 3.5.1 [SG 1] Product or product component solutions are selected from alternative solutions

When choosing potential TMS workflow solutions or connectors to other GMS modules, there are key features to look for when looking for alternative solutions.

Perris (2013), who is an experienced localisation professional and technology manager, in his Top 10 workflow features has extremely well summarised the key requirements which should form part of the solution, namely:

- Project Creation templates
- CAT tools compatibility
- Dedicated Review Portal
- File Filters designer
- Linguistic asset management

Typical solutions to have in mind when designing a workflow in TMS involve the TMS platform itself and any possible connectors to transport the content in and out of the TMS. Such connectors may be simple batch files that allow to push or pull files by executing custom-built tools using TMS API in a command line.
After enumerating the possible solutions for the particular workflow, the workflow engineer liaising with the relevant project stakeholders, needs to select the product component solutions based on selection criteria. (CMMI Institute 2010)

Selection criteria would typically address costs (e.g., time, people, money), benefits (e.g., product performance, capability, effectiveness), and risks (e.g., technical, cost, schedule). (Trinity Management Consultants Limited 2011).

In addition to the above, establishing selection criteria in our case should consider several other components specific to automation and solutions comprising of multiple modules. I will detail those in the remaining part of this SG.

SaaS based solutions, either internal or external to the LSP, offer the advantage of not having to worry about upgrades. Dedicated support teams are in charge of testing and deploying upgrades ensuring compatibility so the solutions stakeholders don’t have to worry. That being said, a good word of advice for SaaS support teams would be to:

Test thoroughly before you deliver us a fix or a solution […]. Another enterprise-strength need on the SaaS side is for reliable fail-over systems and backup.

(Adobe, EMC and other enterprise GMS customers 2011)

Naturally, SaaS solutions developed internally have a critical advantage in that the support teams are much more effective in catering for the LSP use-cases.

It is best to choose a combination of solutions that offer the least amount of customisation:

We need a GMS to offer a wider range of built-in features that are configurable via the user interface, without customizations. Having to customize filters rather than configure them is really a pain in the neck. It introduces significant time delays into projects and is costly. While filters may be nicely configurable on the documentation side, our experience with GMS software filters is that a lot of customization is always required, even though the software file types are very standard. Unfortunately, the more we have to customize, the more testing is required, and the more challenging it is to upgrade.

(Adobe, EMC and other enterprise GMS customers 2011)
It is crucial when evaluating solutions to look ahead. Breaking down the problem, as stated in the requirements, is often a necessity if only to get the users on-board, but the workflow engineer must consider and allow room for improvement:

Initially the focus will be on the user, but increasingly this will be automated combining both unstructured and structured information to truly inform decisions and initiate events.

(Cameron 2011)

Interoperability is key to make it easy to ensure maintenance and future proofing:

Ensure interoperability among different system components by real support for open standards and industry collaboration.

(Adobe, EMC and other enterprise GMS customers 2011)

3.5.2 [SG 2] Product or product component designs are developed

After choosing solution components in SG 1, the final solution needs to be developed. During this process, the workflow engineer should be able to seek the help of support staff as defined by Tools Components/Technician Perspective in the Zachman framework, chapter section 2.1.

Modelling the workflow and all the solution components may be needed if the process is complex using UML or BPMN process tool such as Microsoft Visio or Bizagi/Yasper. Most TMS have workflow designers components that allow workflow engineers to design workflows in a UML inspired graphical interface:

The Workflow Designer should offer a customisable library of human and automated steps and the possibility to have more than one outcome to each workflow step (e.g. Pass/Fail, DTP/no DTP etc.)

(Peris 2013)
It is necessary to look beyond the TMS and also include the various modules from the GMS or CMS it will connect to.

The CMS is designed mainly to help author and manage content. The GMS is designed to help manage the localisation process. The following steps are generally carried out while automating the localisation process, and in the following sequence:

1) Detecting any change in the source content
2) Creation of the jobs for the translators (This includes the updates source content along with the relevant meta-data)
3) Use the appropriate filters to extract the data
4) Segment the data appropriately for large projects
5) Leverage any fuzzy matches
6) Get the approved quote
7) Distribute the work accordingly
8) Perform Translation
9) Review
10) Functional Testing
11) Complete the job
12) Send it to the person managing the multi-lingual content in the enterprise
13) Billing and collecting
14) Job archival

(Lefman 2013)

The workflow engineer should also consider testing broadly beyond the requirements to plan for future expansion based on the organisation standard set of practise. For example:

We had built contingency at every turn, we had tested many times over and had even negotiated some additional time with the client, but reality has a tendency to be different than the lab. [...] And while we tested quite a few file formats during the preparation phase, we didn't test graphics files as they had never before formed part of this project's requirements.

(VistaTEC 2013)
When designing the workflow itself in the TMS, one of the most extensive tasks is to adapt currently automated tasks outside of the TMS in the current process to their counterpart in the TMS. It is not at all inconceivable that many of such tasks (i.e. QA) use a combination of scripts, standalone tools and macros to achieve their intended objective. TMS scripts use specific programming languages (i.e. C# or Python) and only offer access to certain code libraries. As a result, this most likely will involve developing custom scripts or using an existing script available in the TMS scripts repository, if your company has such repository in place.

Especially crucial for complex workflows, the workflow engineer should maintain a technical data package. This will help during the development and implementation phase, but also beyond, during the maintenance phase by the project engineers. Such documentation should be kept up to date throughout the localisation project lifecycle to record essential details of the TMS workflow design, organising its content using the workflow architecture:

One of the useful things to come out of this was a document listing all the potentially tricky situations that might be encountered. This is something we continue to maintain as we probe the tool further, and is one of our first ports of call when we roll out a new workflow, or new file types.

(VistaTEC 2013)

When having to design TMS components, interfaces such as the input and output of the TMS should be carefully configured. The API documentation, if any, should offer keys to connect to or from the TMS workflow.
3.5.3 [SG 3] Product components, and associated support documentation, are implemented from their designs

When implementing the solution components, it is good practice to follow the steps below from GlobalSight Eight Steps to a Successful TMS Rollout by Rock (2010):

- Relieve the biggest pain first
- Test before deployment
- Pace the roll-out
- Address Issues – fast

Finally, like any software solutions, it is necessary to develop and maintain the end-use documentation (CMMI Institute 2010). Despite the intended use of such documentation, along with project stakeholders training, which is to enable them to execute the workflow, the following can be said about good documented processes:

Good processes allow good people to think forward and apply themselves in more value-added ways than in reinventing the routine work every time they need to perform it. Good processes also allow the good people to unload busy work to less experienced people while they go off applying their experience to new ideas and improved performance leaving the routine stuff to people who can follow a process to uphold the status quo performance.

(Entinex, Inc. 2013)
Chapter 4: Well-handledness in localisation workflows

4.1 Petri nets

The concept of workflow well-handledness is tied in to Petri nets.

Petri Nets are a very simple mathematical model. It is a graphical language, yet the semantics are clear. There’s essentially just one rule of action, which is the rule of how transitions are enabled and fire.

The Petri Net model has been around for many years and has been used extensively in simulation and verification of network protocols, a subject that is very like workflows in many ways. Therefore, it comes as no surprise that the concepts used in workflows maps onto Petri Nets in a very straightforward way.

([project-open] 2012)

We can consider, in the remainder of this chapter section, the following definitions according to TU Eindhoven and Deloitte (2005) in Yasper user guide:

4.1.1 Workflows

Workflows are processes in which individual cases (represented by tokens) flow from a fixed starting point to a fixed end point; if such a net can handle such business cases indefinitely and concurrently without ever locking up or amassing tokens anywhere, we say it completes properly.

4.1.2 Net completion

When a cased token arrives at a collector, which is a transition with one or more case sensitive inputs and without sensitive outputs, we say the collector collects the case. If no other token for the case remains, we say the case completes. A model in which every possible generated case is guaranteed to complete has proper completion.
4.1.3 Net well-handledness

In a net, if all branch points form pairs of splits with matching joins of the same type, it is called well-handled.

In well-handled nets, the only form of parallelism is completely hierarchical: a process can start multiple concurrent processes and wait for all of them to finish, but communication or synchronisation between different sub processes is impossible.

In practice, a large class of processes can only be modelled with nets that properly complete, but are not well-handled. This situation can arise as soon as the constraints on the order of process steps are anything other than a choice between fixed paths.

In Yasper, the user sees the tokens flowing through the net and spots deadlocks or bottlenecks where they arise. Simulation turns out to be an extremely convenient tool in designing correct non-well-handled nets, and appears to catch nearly all of the modelling errors that arise in practice.

If designing well-handled nets, they always complete properly so it is a good design principle to stick to neatly bracketed split-join pairs whenever possible. Doing so makes problems easy to spot.

4.2 Well handledness in TMS

The following Petri nets properties as defined by TU Eindhoven and Deloitte (2005) apply to most if not all TMS workflows:

- all branch points form pairs of splits with matching joins of the same type; if a net has this property, it is called well-handled

- Workflows are processes in which individual cases (represented by tokens) flow from a fixed starting point to a fixed end point. If such net can handle such business cases indefinitely and concurrently without ever locking up or amassing tokens anywhere, we say it completes properly.

Inspired by one of the real-life processes in Lionbridge, the modelling of a State of the art Lionbridge process using Petri nets in Appendix C shows a process that clearly has the above properties, with little human intervention.
Indeed, the file HO, HB, pre-process and post-process steps are automated with translation and review as the only recurrent human steps. An engineer is only involved if errors are detected during the xliff qa check conditional activity as part of the post-processing workflow sub-level.

In this example, the translation workflow sub-level is a direct illustration of the well-handledness property. Indeed, the translation sub-level workflow for all three languages run in parallel, independently from each other and with no communication or synchronisation between each other.

Petri nets transitions can be mapped to activities in TMS. Main activities in TMS are either human, system or flow control activities as it can be seen on the Lionbridge TMS Workflow Designer toolbox in Appendix D.

Additionally, workflows in TMS can be summarised using the following rule: a job has one or more target languages translating one or more files. This is why there are essentially three levels where activities can be run: job level, language level and at the lowest level, work item (or file-based) level.

Well handledness is strictly inherent to the above where matching pairs of branching and merging are required to go one level up or one level down, as we will explore in more detail in Chapter 5, during the Lionbridge TMS case-study.

Activities also abide by the concept of well handledness which can be easily observed with activities like chunking and reassembling:
[Figure 4.1] 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).

(Filip & Conchúir 2011)
Resource predictability is key at the system level, where all workflow cases need to be known in advance to minimise the risk of problems. As a result, only well-handled nets are supported. Providing even limited support of non-well-handled workflows would require allowing potentially unlimited resources. If such workflow was allowed, generated cases would not be guaranteed to complete properly, which contradicts the very definition of a workflow according to TU Eindhoven and Deloitte (2005).

With the above in mind, TMS allocates system resources for the workflow activities when starting a new job. It does so generally by making a copy of the workflow template associated during job creation. Activity settings may be the only element that is not cached in advance and loaded at run-time on an as-needed basis, as they are not unpredictable in terms of resources consumption. Subsequently, it is possible to understand common TMS inherent limitations:

- Flow control can only go forward, not backward to an already executed activity,
- Loops are not permitted, and
- The workflow engine can only navigate along the cases defined by the workflow. If an alert for a given activity is thrown, the system cannot redirect to a particular user, as it would otherwise mean creating an "unplanned activity", so a user has to action it in place.
- Sub-level processes such as language level or file level which execute in parallel are black boxes where “communication or synchronization between different sub processes is impossible.” (TU Eindhoven and Deloitte 2005). As a corollary, conditional activities can only look at the current level, using conditionals such as file extension, file name, file content or a previous version of a file from a previous activity.

All that being said, it is concerning that TU Eindhoven and Deloitte (2005) have claimed that “in practice, a large class of processes are not well-handled, or in other words, the order of process steps is not a fixed path.” We shall explore if that claim is true in the context of localisation processes, by exploring the concept of workflow patterns.
4.3 Workflow patterns

A process-aware information system (PAIS) is a software system that manages and executes operational processes involving people, applications, or information sources on the basis of business process models.

(Aalst & Stahl 2011)

In process-aware information systems various perspectives can be distinguished:

- The control-flow perspective captures aspects related to control-flow dependencies between various tasks (e.g. parallelism, choice, synchronization etc.)
- The data perspective deals with the passing of information, scoping of variables, etc.,
- The resource perspective deals with resource to task allocation, delegation, etc.
- Finally the patterns for the exception handling perspective deal with the various causes of exceptions and the various actions that need to be taken as a result of exceptions occurring.

(Workflow Patterns Initiative 2007)

As we have seen in Chapter 1, localisation processes are defined and follow what seems to be in theory a fixed path. In practice, during any localisation workflow execution, there may be unplanned factors that alter the normal path of defined process steps such as below:

- Change of scope: new or removed languages or files, revised deadline, revised instructions or context files
- Resource unavailability: a translator is unavailable, a task is unclaimed by the user
- Deadline Expiry: deadline for a work item is reached
- Work Item Failure: a file gets corrupted, job is cancelled, failure of underlying hardware, software or network (i.e. see applications detailed in Table 2.1.)
Those unplanned factors may occur at any stage of a workflow and should ideally be handled correctly in any TMS. We will examine workflow patterns from the control-flow and exception handling perspectives\(^3\) to determine to which extent are non-well-handled processes supported or not. Our case study of Lionbridge TMS in Chapter 5 will show that this support is rather limited in today’s TMS.

\[^3\] Due to limited time, we will focus only on the Control-Flow Patterns and the Workflow Exception handling patterns. As a result, the Workflow Resource Patterns and Workflow Data Patterns perspectives are out of scope of this paper.
Chapter 5: Case study of Lionbridge TMS

This chapter will evaluate Lionbridge TMS around process well-handledness using workflow patterns as defined by the Workflow Patterns Initiative (2007).

The evaluation of Lionbridge TMS\(^4\) will be based on extensive testing of the platform as well as drawing from the publicly available Lionbridge TMS™ Help for End-users by Lionbridge (2013).

5.1 Background

This system is available as a SaaS service accessible inside and outside the Lionbridge company firewall. All you need is to log in on https://tms.lionbridge.com. The Help section is also publicly available at http://tmshelp.lionbridge.com.

Implementing a program onto TMS involves high level steps documented in Figure 5.1.

![Figure 5.1: Internal TMS Template Implementation (Lionbridge 2013)](image)

\(^4\) Due to limited time, only Lionbridge TMS will be examined. Other mainstream industrial TMS such as SDL TMS / SDL WorldServer and GlobalSight won't be discussed in this paper.
In practise, generally the lead engineer explores the benefits of automation for a given account, either tasked by the PM, or as part of continuous improvement work practises. If TMS is deemed a realistic approach to the problem, the lead engineer then works with a dedicated Lionbridge TMS workflow engineer in-house expert. The workflow engineer, with the input from the lead engineer and other stakeholders, will then build the most suitable workflow to automate as many steps as possible (file preparation, log generation, user assignment, conditional activities, etc.)

Skipping the implementation part documented in detail in Chapter 3, the workflow is now live. A job is created containing various files that need to be translated and a Workflow Configuration is used as a template containing all the steps necessary for the job to complete. Language scope, project title, deadlines are also specified at that stage.

TMS may not be the only part of the solution. Indeed, depending on the account, there are also various tools or batch scripts that may be needed to work with TMS. For instance, it may be necessary to automate the creation of jobs as soon as the client sends HO files to a particular location or even automate the HB (renaming, file collating, or delivery back to the client).
The following steps are typical together with the TMS progression example in Appendix E:

1. When a job is started, the first step may consist in running scripts or file preparation activities to get the files ready for translation (i.e. search and replace using regular expression, encoding conversion, etc.)
2. Then, the files are marked up and converted to Bitext formats such as XLZ (proprietary ZIP container based on XLIFF) or RTF, by connecting to an external application called Translation Workspace.
3. Files can then be pre-translated.
4. Analysis logs showing the leverage statistics from the TM are automatically generated and visible for PM and translator.
5. The XLZ files (or file splits based on word count) are sent to a translator or a pool of translators to work on.
6. After translation, there may be an In-context review stage (ICR) where files get automatically back converted from XLZ or RTF to their original format.
7. Automated script condition activity to check if XLZ files are sound.
9. Optional human engineer QA steps (i.e. Full HTML or XML QA).
10. Final delivery kit creation.

Some accounts may therefore be almost totally automated with the exception of translation and review tasks and ad-hoc engineers QA such as shown previously in the State of the art Lionbridge process using Petri nets in Appendix C.

During the project duration, the PM uses a Planning view to check the live progress of any given jobs and corresponding human tasks status. There is also the possibility to download reports in Microsoft Excel format.
A few benefits to take away:

- Automating repetitive, time consuming, error-prone and boring tasks such as file pushing and emails is one of the great benefits of such system.

- Visibility for the PM as to where the files are at any given time to better manage client expectations.

- Automation allows introduction of very aggressive Service Level Agreement (SLA) (i.e. 24 turnaround time) on accounts which can be totally automated (i.e. XML source files structurally consistent).

- Less irrelevant carbon copied emails distracting project stakeholders.

- The engineers’ expertise is focused on mindful maintenance of the system rather than performing repetitive mindless tasks.

- Less problems caused by the translators: they are constricted by the system check-in rules which will not proceed until the file set with the same filename as during check-out have been checked back in.
### 5.2 Terms and basic definitions

In order to evaluate Lionbridge TMS, it is important to understand some key workflow concepts. All terms have been extracted from the TMS Workflow Designer manual (Lionbridge 2013) except those marked with an asterisk. They have been separated from the overall glossary for the paper for clarity purposes.

<table>
<thead>
<tr>
<th><strong>Activity</strong></th>
<th>An Activity is an action to be performed either by the TMS system or a person. One step in the workflow includes one Activity (cf. Appendix D for a full list of TMS activities).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activity Setting</strong></td>
<td>The configuration of one workflow activity (for example Analysis, Filtering) saved as a setting with parameters that can be included in a Workflow Configuration.</td>
</tr>
<tr>
<td><strong>Branching</strong></td>
<td>Branching divides a workflow into two or more parallel branches, each of which execute concurrently.</td>
</tr>
<tr>
<td><strong>Conditional jumping</strong></td>
<td>Conditional jumping uses decision points set into the workflow to control the flow of activity executions. Conditional jumping skips steps when the condition is met.</td>
</tr>
<tr>
<td><strong>Freeway</strong></td>
<td>Lionbridge cloud-based Translation Management Portal.</td>
</tr>
<tr>
<td><strong>Job</strong></td>
<td>A Job is a set of files in a project that flows through the steps defined in a workflow configuration, resulting in a series of Tasks. A job has a start and end date, a source language and one or more target languages. A Job contains the tasks in all locales. A user sees only tasks assigned to her/him. When a Job is created, a Workflow Configuration is used as a template.</td>
</tr>
<tr>
<td><strong>Lock Frequent Segment</strong></td>
<td>Function to lock the frequently repeated segments in a batch of XLZ files before the files are otherwise processed.</td>
</tr>
<tr>
<td><strong>Logoport</strong></td>
<td>Lionbridge TM stored on online central database.</td>
</tr>
<tr>
<td><strong>Merging</strong></td>
<td>Merging unites two or more parallel branches into a single subsequent branch in which the thread of control is passed</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Project</td>
<td>The container of jobs received from a customer with a specific TMS configuration (workflow templates, project members). Inside a project there are jobs that include job items, which include tasks.</td>
</tr>
<tr>
<td>Step</td>
<td>A step is one instance of an activity chain for a Job, a Job Item or a Work Item.</td>
</tr>
<tr>
<td>Task</td>
<td>A Task is the action (for example translation, review) performed in one workflow step by an End-user to a file to be localised, or a generic task, which the Lionbridge TMS® system takes care of (for example analysis, file conversion).</td>
</tr>
<tr>
<td>TMS*</td>
<td>Lionbridge TMS</td>
</tr>
<tr>
<td>Translation Workspace*</td>
<td>Lionbridge cloud-based translation productivity platform. It also comprises a set of tools to prepare files for translation called Translation Workspace Tools.</td>
</tr>
<tr>
<td>Work Item</td>
<td>A Work Item is one of the Job files for a given source and target language.</td>
</tr>
<tr>
<td>Workflow Configuration</td>
<td>A Workflow Configuration is created by combining a group of Activity Settings with a Workflow Template. The configuration is used for creating new client specific projects into the Lionbridge TMS.</td>
</tr>
<tr>
<td>Workflow Template</td>
<td>A predefined Workflow Template including different localisation project steps for a Job from word count analysis to delivery. The activities can be further configured by the Project Administrators.</td>
</tr>
<tr>
<td>XLZ*</td>
<td>Logopoort Zipped XLIFF file format which contains a proprietary skeleton file and custom flavour of XLIFF.</td>
</tr>
</tbody>
</table>
There are three fundamental levels in any given workflow visible in Lionbridge TMS Workflow Designer as soon as a workflow engineer creates a new workflow template (cf. Figure 5.2).

**Job sequence:**
Set of activities to execute for entire job.

**Job item sequence:**
Set of activities to execute per language.

**Work item sequence:**
Set of activities to execute per source file.

*Figure 5.2: Blank Lionbridge TMS Workflow Designer workflow template*
5.3 Evaluation using workflow patterns

Workflow Foundation is the name of the Microsoft library on top of which Lionbridge TMS workflow engine is built. All research in this paper was performed on WF3 (Workflow Foundation 3) on TMS 3.5.

We will evaluate Lionbridge TMS against the workflow patterns from the various perspectives defined in chapter section 4.3, using the evaluation techniques defined by Workflow Patterns Initiative (2010). The following rating method has been used:

For each product-pattern combination, we checked whether it is possible to realize the workflow pattern with the tool. If a product directly supports the pattern through one of its constructs, it is rated +. If the pattern is not directly supported, it is rated +/- . Any solution which results in spaghetti diagrams or coding is considered as giving no direct support and is rated -. Note that a pattern is only supported directly if there is a feature provided by the graphical interface of the tool (i.e., not in some scripting language) which supports the construct without resorting to any of solutions mentioned in the implementation part of the pattern.

(Workflow Patterns Initiative 2010)

TMS needs to handle process deviations. "It is only possible to specify handlers for expected types of exception." (Workflow Patterns Initiative 2007).

Exception patterns take the form of tuples comprising the following elements:

- How the task on which the exception is based should be handled;
- How the case and other related cases in the process in which the exception is raised should be handled; and
- What recovery action (if any) is to be undertaken.

(Workflow Patterns Initiative 2007)

Appendix G shows the result of the evaluation of TMS using Control-Flow workflow patterns whereas Appendix H evaluates TMS against the Exception handling patterns.

When evaluating exception handling in commercial offerings, the Workflow Patterns Initiative (2010) insisted on the “lack of support for the resource perspective in current commercial products.” Lionbridge TMS seems to be in disagreement with this statement.
However, regarding the "minimal support for external triggers and constraint violation management" in commercial offerings, this statement applies to TMS, which offers no support for external triggers (cf. chapter section 5.4.5.3) or constraints violations.

In conclusion of this survey, exception handling in TMS seems acceptable, definitely more sophisticated than the Control-Flow patterns support which is limited to only basic patterns. This priority is understandable in such system where if things go wrong during production, the business needs to know about it as early as possible to apply corrective action.
5.4 Limitations

The following limitations were compiled during the survey of TMS. They are categorised using patterns defined by Workflow Patterns Initiative (2007).

5.4.1 Abstract syntax limitations

5.4.1.1 No Vertical Modularisation

Customising multiple levels in workflows is not supported. Therefore, it is not possible to “increase the understandability of large process models by "hiding" process details into sub-levels” (Workflow Patterns Initiative 2012).

Workflows sub-levels would be also quite useful to define smaller workflows as black boxes that one can reused in other workflows to save time and make workflows easier to read and maintain.

For example, in Lionbridge, steps in Figure 5.3 are typical involving:

- Translation
- Engineering (ENG)
- Language Coordinator (LC) to check Language Toolbox (LTB) reports
5.4.2 Control-Flow limitations

5.4.2.1 Limited support for human decision points

With TMS, a PM can approve or reject a project by the mean of the Approval human activity where the user can either mark a given task as Passed or Failed during workflow execution as it can be seen in Figure 5.4.

Depending on the Passed or Failed options chosen in the user interface, the activity options in the workflow design level determine where to jump to in the workflow (cf. areas marked with red and green diamonds in Figure 5.5).
TMS only allow this level of control once a workflow is running. It would be useful to have more decision-making power in the system. For instance one could envision, as part of a single PM activity, customisable options (aside from Passed and Failed) to control the workflow branching (cf. Multi-Choice Control-Flow pattern in Appendix G).

Typically, once a PM hands-off a localisation kit to the engineering team they have to provide information, as part of a template, which is documented in Appendix I.

5.4.2.2 No run-time control to skip steps in workflow

A step in a running workflow can only be skipped if an error concerning this step is thrown during workflow execution. The user cannot skip an activity of his or her choosing. Skipping steps could be very valuable. For instance, if certain languages or files need to be skipped after a job has been started.

Sometimes, certain customers require that certain files sent by Freeway to TMS pass through untranslated back into Freeway and sent back to their CMS tool. The reason is that some clients may realise that files should not be translated when their CMS have already handed over the files to Lionbridge via Freeway and they are in translation stage here. For some clients, this is a common request. These requests should supersede any current task anywhere in the workflow (i.e. translation or review).

Currently, this is only possible if no transformation script exists in the workflow and requires that the file be pushed through manually at each human step.
A typical example for an eligible workflow would be:

1. The Language lead un-assigns any task accepted by the translator or the reviewer.
2. The engineer passes through files at each stage of the workflow toward the last step. This takes about 5min time where the engineer has to assign to self, check-out and check-in files during translation stage, marking any QA tasks detecting untranslated files as Passed and Complete, assigning to self and marking the Editing task as Complete.

This is a highly inefficient file pushing job.

A possible development solution would be to have an option on a job to Pass-through file(s) untranslated where a user would be redirected to select the files and languages for that job that needs passed through untranslated. If implemented, a PM could run this task which would avoid having to wait on the engineer to perform the task.

Another more ideal solution would be to change the customer process and agreement with them in such way that if all files for a given job for all languages should not be translated, the PM would follow the approach below:

1. Cancel the job in TMS. If tasks are in progress by external resources, force un-assign.
2. Deliver source files in Freeway manually.

Otherwise, push the relevant files and languages through TMS as per existing file pushing process aforementioned.
5.4.2.3 No custom branching

TMS only supports sequential activities within a given branch, that being Job Item (language-level) or Work Item (file-level). Therefore, it is not possible to branch the workflow at Job Item level as below yielding two simultaneous activities *Translation* and *Reference*, where the translator receive two tasks at the same time to check out reference materials and file for translation:

![Diagram](image1)

*Figure 5.6: Custom parallel branching concept*

The problem is that TMS only supports sequential activities within a pre-defined branch, so only the below linear approach is possible: translator checks out reference materials during *Reference* activity, mark task as complete and then receive a task corresponding to the *Translation* activity.

![Diagram](image2)

*Figure 5.7: Sequential activities in TMS*
5.4.3 Resource limitations

5.4.3.1 No run-time loop with historic based allocation

Let us consider the scenario below where a PM wishes to reassign a file in review step back to translation.

A translator translated all segments in a file with no issues except that further work was still needed and the file was prematurely checked in with the task marked as Complete. The file went straight to review task as the conditional script activity in-between used for QA checks returned True because the file had all Bitext segments translated. The PM would like to return the file back to the translator for further work.

The workflow cannot go back to the completed translation step, as per control-flow patterns in Appendix G.

Let us consider another scenario where a translator accidentally claims a task and instead of un-claiming, checks-in the file, preventing from un-claiming again.

As a workaround, it is possible, and actually good practice, to have a script ahead in the workflow that detects untranslated segments and send to an engineer where the following QA steps are performed:

1. Check if translator checked in the untranslated file,
2. Mark task as complete to progress to script step,
3. Check out the files, email to PM to send to translation, and
4. Check in translated files to resume workflow.

The support for loops at run-time to resume the workflow from a previous activity would solve the problems in both scenarios. More realistically, an alternative solution would be to offer more flexibility in overriding translators controls and thus able to reassign a task at any stage (such as if file has already been checked in) before the task is marked as Complete.
5.4.4 Data limitations

5.4.4.1 Limitation in post work log generation

After translation is complete, billing is generally accomplished outside of TMS. However, let us consider a scenario where multiple translators work on the same project. They would save their work to a common TM and thus benefit from each other’s work during the translation process.

The most accurate compensation for this work should not be based purely on the quote, which was generated before starting the work, but instead on the TM leverage statistics. If TMS supported it, it is likely that the PM would want to pay a lower amount for any new 100% exact matches, fuzzy matches (less than 100%), that would have been added in the TM by other translators working on the same project, from the time the quote was issued.

Of course this approach would have to be agreed with the translators or MLVs. The point is that to pay in such way would require the TMS to leverage this statistics from the XLZ files. However, the only analysis activity in TMS generates statistics based on the TM content and not the fuzzy match value in the XLZ Bitext files themselves.

5.4.4.2 No Correlated files support for Human activities

Certain system activities can refer to a correlated file from a previous step (i.e. a script which take two inputs: the file at current step and a correlated file, from a past activity).

By default the files are always copied from the previous activity to the current step. Using the Correlated Step property allows the workflow engine to give read only access to a version of the files from a preceding activity specified in the Correlated Step.

(Lionbridge 2013)

Only single versions are available at any Human activities. For example, it is impossible to retrieve the source file from a previous step (i.e. to use as context during translation), only the current version of the file (i.e. Bi-text version during translation step) is available.
The only way is to have a separate file considered as a context file using a *File Type Mapping* setting but it is impossible to have a file format mapped to more than one file type. For instance, you cannot set a file extension to be both a localisable file and context file, you have to choose.

The best solution would be to allow to choose both context and localisable for a given file extension. Another proposed solution although specific to this use-case would be to insert an option to download source files next to the option to check out files in the Tasks view during a Human activity.

### 5.4.4.3 No source/target pair data integrity checks

It is not currently possible for TMS to detect structural discrepancies between source and target files. However, at any point in a workflow with human activities, between check-out and check-in, things can go wrong.

For instance if an XML file has more or less nodes in the target file compared to the source, you would want this flagged to an engineer to check and fix the file.

Another practical example would be that TMS relies on unique ID affixed to file names during check-out. This ID is necessary for binding the files back during check-in. Let us suppose that two or more files are checked out by a translator who then mistakenly swaps the name of one file for another. In such a scenario, you can see how things would go wrong and would fool the system. The workflow branch would progress as normal without throwing any exceptions.

TMS needs to offer built-in and custom options to compare the source and target file content at a structural level against a set of rules. The structure of the file would generally be defined during the conversion to Bitext format.
5.4.5 Exception handling limitations

5.4.5.1 Basic support for task errors handling

Corrupted files submitted by translators can be caught by a conditional activity to determine whether or not QA is needed. Such activity (i.e. XLIFF_QA_Script in Figure 5.8) would throw an alert, usually with a Traceback error message to help troubleshooting.

Such alerts are visible in the project dashboard or relevant stakeholders subscribed to the project RSS feed can be notified in real-time.

![Figure 5.8: Failed activity alert in TMS](image)

Unfortunately, if there is an issue with the files, there is no possibility to re-assign such failed activities to a specific user. The files need to be corrected by checking out the files and working outside of TMS.
In Figure 5.8, the files checked in by the translator were corrupted. It would be ideal to just be able to re-assign to the translator with a comment asking them to check in the correct files but unfortunately, as we saw in control-flow patterns in Appendix G, TMS just does not allow run-time workflow redirections.

Here are the current PM resolution steps:

1. Ask translator to overwrite all XLZ in attached ZIP by the correct files, making sure the name containing the unique identifier is as per files in that archive,
2. Go to Status view and in the Alerts tab select all the 10 alerts corresponding to the corrupted files, check in the archive sent by the translator and click on Retry as shown in Figure 5.9.

3. In the future, translators need to ensure that the XLZ they are checking in can be opened.

Figure 5.9: Alert control options in TMS
5.4.5.2 *No explicit timeout for external applications*

Sometimes, files go into a limbo state where they do not fail a task and alert anyone, instead they just hang in the system and have to be manually extracted from TMS and processed manually to completion.

This is because TMS passes the thread of control to external applications for certain activities such as *Translation Workspace* application to convert files to bitext XLZ format and said application may not return a timeout, causing TMS to wait seemingly forever.

There should be a timeout task failure exception to alert an assignee to fix the issue and restart the task again so it completes in TMS and runs the remainder of the workflow.

5.4.5.3 *No runtime support for external triggers*

Sometimes, a fix to a file needs to be applied before a file gets delivered to the client, say if it was observed that a script activity broke a file earlier in the workflow execution, but the job does not have any more human activities pending execution to modify the file.

This happens if only system activities remain in the underlying workflow. The problem arises if the workflow is set up to push the files to a content delivery platform at the end as the issue needs to be fixed somehow before the file gets delivered.

Unfortunately, once a job based on a workflow has been initiated, it is not possible to edit the underlying workflow associated with the job as it is a static copy of the project workflow template.
As a workaround, the engineer can insert a break point ahead of the current task by removing the setting of a corresponding activity. This allows the engineer to perform the following troubleshooting tasks:

1. Check out the file from the alert,
2. Apply the fix and check in the fixed file,
3. Retry the activity from the alert to force a retry using the newly checked-in files.

This is not such a good workaround because the workflow settings are not stored at a job level but shared for all workflow instances based on the same workflow template. Therefore, removing a setting actually affects all current and future jobs.

Manual workflows have a key advantage in that they offer full flexibility to halt a workflow at any point in time and allow to backtrack to a previous step or amend the executing workflow structure. As we have seen in the control-flow patterns in Appendix G, this is not supported by TMS.

A few solutions come to mind:

1. Being able to alter the workflow of a running job to add, edit or remove steps.
2. Show all non-human remaining activities in the status view instead of just the remaining human activities and allow to set a pause to any given pending activity on a per file or per language basis, which will trigger an alert when an engineer can check-out, manipulate files and check corrected files back in again.
3. An option to insert a pre-defined manual task to a running workflow such as ENG_maintenance which would just be a human placeholder task that can be assigned to self or another engineer.

TMS needs to offer more flexibility at run-time. Humans make mistakes and it is critical to handle such exceptions and offer a flexible approach involving external triggers.
Chapter 6: Conclusion

It seems that TMS is an obvious solution in the perpetual conundrum in the localisation industry of delivering localised files faster, at an increased quality and reduced cost. That being said, before force feeding template-based TMS workflows to your users, automating every localisation use-case, we should ask ourselves to what extent traditional localisation workflows can accommodate this without breaking? How a flexible TMS can cater to the various types of use-cases that make up sometimes complex and often varying localisation processes?

Workflow patterns and the concept of well-handledness are powerful tools to evaluate TMS in an attempt to answer such questions.

The approach followed in this paper was to start by looking at the current state of affairs in traditional localisation workflows. At first glance, the cycles of Translation, editing and proofreading are well defined, following a waterfall model structure in the majority of localisation processes. However, there is a growing for Agile localisation occurring while content is still being authored. This results in the need for structures that can bend more easily to cater to those changing needs.

In parallel, stakeholders must consider business processes from a bird’s eye view to define precisely where TMS comes into action. TMS is not a monolithic system. TMS comes together with separate modules controlling linguistic assets, translation engine, billing modules and job submission portals to form a GMS. It is also common to have one or more CMS connected to it to constitute end-to-end content delivery mechanisms.

Using the Zachman framework for enterprise architecture, we defined the business model of LSP where TMS would be of value and used the ECM methodology to outline the transformative trend in the content lifecycle that sees TMS rise in popularity.

It became apparent that workflow engineers need to carefully define the role of the TMS from a black box perspective by taking into considerations the applications and stakeholders from the various departments involved in the localisation operations whilst not losing focus on key data points from the other organisational perspectives such as business intelligence and reporting. Expectations are then
managed by defining KPIs to measure TMS performance and in turn its added value.

To pave the way for the actual design and implementation of process migrations to the TMS, specific process areas were outlined using the CMMI process improvement methodology.

Driven by the CMNI methodology and the researched data presented up to this point, we defined a detailed roadmap for TMS design and implementation, focusing on important relevant process areas and Specific Goals (SG) inspired from software development. After carefully gathering requirements, the workflow designed in the TMS has to offer as much flexibility as possible, incorporating enough human tasks to cater for change in requirements, but not more than needed so as to not detrimentally affect schedule, as well as a healthy dose of system tasks to automate repeatable tasks. Just like traditional localisation workflows where the human element is at the core of the workflow, the system must be robust enough to allow time-critical maintenance on running production workflows.

After building the foundations of TMS, it was time to introduce in more detail the appropriate tools to evaluate such system before we use those in our case-study. In other words, we defined the concept of workflow well-handledness and the workflow patterns methodology. In well-handled nets, all branch points form pairs of splits with matching joins of the same type, constricting the flow of information. In turn, the workflow patterns allow us to scrutinise this flow around complementary perspectives, that of control-flow, data, resource and exception handling. Variables such as change of project scope, human resource constraints or work item failure were introduced.

The research in this paper led to a case-study of a workflow patterns evaluation of Lionbridge TMS around the concept of process well-handledness. This survey was guided by the process areas and specific goals defined previously. The case study confirmed the main assumption set forth in the introduction of this paper, namely that control-flow patterns support is rather basic and limited to well-handled use-cases, with a fixed set of well-defined transitions.
This is understandable in such SaaS production platforms where the priority is to ensure a high quality of service for the end-user, rather than giving the workflow designer more flexibility when creating workflows or adapting them from traditional localisation processes. In order to meet their Service level Agreement (SLA), LSPs need to support resource predictability at every turn. All workflow cases need to be known in advance or at least predictable.

In line with this business strategy, exception handling on work item failure, deadline expiry and resource unavailable is on par with the expectations set forth. However, providing support for non-well-handled use-cases should be sought after. It need not be at the expense of system integrity and upkeep but instead, as part of continuous system improvement.

Many limitations have been uncovered during the course of this research after using the system on a daily basis for production tasks. Corresponding solutions to those have been suggested but it is unrealistic to expect the support for non-well-handled use-cases to change in the near future. Rather, improved workflow patterns support may be partially provided, but it likely will be based on well-handled workarounds.

It would surely be interesting to see how other systems such as SDL WorldServer or GlobalSight fare compared to Lionbridge TMS although I expect them to be on a par with one another and also limited in their support for non-well-handled processes.

As a final take away, there is tremendous potential for innovation and growth if industry leaders such as Lionbridge looked towards:

- Improving the support of workflow patterns in their TMS offerings, and in parallel
- Implementing workflow recommendation wizard in their workflow designer tools, in line with the current research led by Adam Morera Mesa at the Localisation Research Centre (LRC).
Bibliography


Rock, H. (2010), 'Eight Steps to a Successful TMS Roll-Out', *GlobalSight* [Online], available:


Workflow Patterns Initiative (2007), 'Exception Types', *Workflow Patterns* [Online], available:  

Workflow Patterns Initiative (2007), 'Patterns', *Workflow Patterns* [Online], available:  

Workflow Patterns Initiative (2010), 'Commercial Products Evaluation', *Workflow Patterns* [Online], available:  

Workflow Patterns Initiative (2012), 'Pattern 4 (Vertical Modularization)', *Workflow Patterns* [Online], available:  

Zachman International, Inc. (2011), 'About The Zachman Framework™', *Zachman International* [Online], available:  

Zachman International, Inc. (2012), 'New Zachman Framework™ Announced!', *Zachman International* [Online], available:  

Zaki, A. (2013), 'The Challenges of Agile Localization', *Globalization and Localization Association* [Online], available:  
http://www.gala-global.org/node/84763 [accessed 15 August 2013].
Appendices

Appendix A: Typical localisation process using BPMN

Created using Bizagi Process Modeler version 2.4.0.8. The overall model has been split vertically into two parts for easier formatting in this paper.
Hi David,

For my dissertation (see a few PowerPoint slides attached that I had to present at the UI in May), I chose to focus on TMS and in particular how they handle exceptions, how flexible they are at handling various types of processes. I was wondering if we could discuss to what extent I’d be able to use my knowledge of Lionbridge TMS in this paper and if I could use the system (web interface and workflow designer) to run experiments in a non-production sandbox (see attached PDF for recent draft report outlining the structure of my paper and points I plan to address such as Lionbridge TMS).

I was planning to use the result of my research such as below to submit to TMS team for ideas on improving the system:

<table>
<thead>
<tr>
<th>Exception scenario</th>
<th>Current handling</th>
<th>Suggestion for improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>If a translator accidentally claims a task and instead of unclaiming checks in the file, preventing from unclaiming again.</td>
<td>Have a script ahead in the workflow that detects untranslated segments and send to ENG for QA. Check if translator checked in the untranslated file and then mark task as complete to progress to script step. Then, check out the files, email to PM to send to translation and check in translated files to resume workflow.</td>
<td>More flexibility in overriding translators controls and being able to reassign a task at any stage (such as if file has already been checked in) before task is marked as complete.</td>
</tr>
<tr>
<td>Fix issues in target files before delivery.</td>
<td>For any “In progress” jobs, it is possible to fix issues by removing a setting in the workflow. The best is to remove the PostScript setting so as to be able to fix any issues in the backconverted XML files. Any current jobs will now always fall upon reaching that step. You may have then necessary hook to check out the backconverted files, apply a fix in the target, perform the same manual operations as that of the “PostScript” and then check the files back in, stopping activity to move to the next step in the workflow and thus complete.</td>
<td>Once a job based on a workflow has been initiated, it is not possible to edit the underlying workflow this job is based on. A few solutions: 1. Being able to alter the workflow of a running job to add/edit/remove steps 2. Mark a break point (like in code for compiler) at any place in a workflow to insert a pre-defined manual task to a running workflow such as ENG files which is just a human treadmill task assigned to self to be able to check out, manipulate files and check back in again.</td>
</tr>
</tbody>
</table>

I was then thinking to use the questions/scenarios defined during the Lionbridge TMS analysis as basis for questions/research points for potential interview/sampling of other tools such as WorldServer, etc. to draw comparisons of a few TMS.

Also, for that dissertation, to what extent would I be able to draw/quote from my 6 sigma training materials or maybe it would be best to use public sources such as https://en.wikipedia.org/wiki/Six_Sigma

Thanks,

Nicolas Martinez
Ingeleer Localisation / Localisation Engineer Grade 3, G1T
Lionbridge Technologies, Inc
3 West-Pier Business Campus, Dun Laoghaire, Co. Dublin
Appendix C: State of the art Lionbridge process using Petri nets

This model has been created using Yasper version 1.0

Main workflow
Workflow sub-levels

automation pushes HO to TMS:

[Diagram of workflow sub-levels]

automated file prep:

[Diagram of automated file prep]

fr-fr translation (same for de-de and zh-cn):

[Diagram of fr-fr translation]

fr-fr review (same for zh-cn):

[Diagram of fr-fr review]

engineer qa:

[Diagram of engineer qa]

automation pushes HB to FTP:

[Diagram of automation pushes HB to FTP]
Appendix D: Toolbox in Lionbridge TMS Workflow Designer
Appendix E: TMS progression example (Lionbridge 2013)
Appendix F: Pivot process in TMS (Lionbridge 2013)

The Pivot Process is only executing for languages requiring Pivot language translation.

The Pivot Process will start when the Common Process of the Pivot language is completed.

The Source Language for the Job Item will be changed to the Pivot language.

The Common Process is executed for all languages.

The Common Process for language requiring Pivot language will be executed after the Pivot Process is executed.
## Appendix G: Control flow patterns TMS evaluation

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Rating</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>+</td>
<td>Directly supported by arcs (drawn as lines from top to bottom) connecting steps (cf. blank template in Figure 5.2: Blank Lionbridge TMS Workflow Designer workflow template).</td>
</tr>
<tr>
<td>Parallel Split</td>
<td>+</td>
<td>Supported by branching in sequence activities such as Job Item Sequence or Work Item Sequence levels (levels which can also be set for any activities in the Level flow control property). Branching is also allowed by Condition or Script Condition flow control activities, although in those cases the split itself is not displayed in the Workflow Design pane.</td>
</tr>
<tr>
<td>Synchronization</td>
<td>+</td>
<td>Supported by the underlying workflow engine. The next activity will only be reached once all of the activities of all incoming branches have completed.</td>
</tr>
<tr>
<td>Exclusive Choice</td>
<td>+</td>
<td>Supported through the condition construct modelling a binary decision. TMS can only jump to one step out of a choice of two steps based on the result of a Boolean function. This function and On Failed or On Succeeded properties are defined in the Condition or Script Condition flow control activities.</td>
</tr>
<tr>
<td>Simple Merge</td>
<td>+</td>
<td>Supported by merging. All parallel split branching constructs come with a matching merging pair, thus only supporting well-handled nets. The merge itself is not displayed in the Workflow Design pane other than, for sequence activities, by the borders delimited its boundaries. For example, a script can be used after a DeliveryKit activity to merge individual language files into a single file. For example, Excel file or XML with multiple languages nodes.</td>
</tr>
<tr>
<td>Pattern</td>
<td>Rating</td>
<td>Motivation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Multi-Choice</td>
<td>+/-</td>
<td>Not directly supported. The condition construct can only model a binary decision as per the Exclusive Choice pattern. As a workaround, use multiple sequential conditional activities, which takes routes of all True conditions (More information in chapter section 5.4.2.1)</td>
</tr>
<tr>
<td>Structured Synchronizing Merge</td>
<td>-</td>
<td>Not supported due to Multi-Choice support where multiple sequential conditional activities don’t execute as parallel branches but in a single branch.</td>
</tr>
<tr>
<td>Multi-Merge</td>
<td>-</td>
<td>Not supported. Although more than two parallel Job Item Sequence or Work Item Sequence branches may execute in parallel, the subsequent activity after the merge only executes after all branches have merged (i.e. a DeliveryKit activity at Job Item level waits for all languages threads to complete to execute).</td>
</tr>
<tr>
<td>Structured Discriminator</td>
<td>-</td>
<td>Not supported. It is not possible to skip an activity running in a parallel branch; although there is some limited level of communication between parallel branches (see Milestone pattern), the execution path is still fixed and thus it still does not allow non-well-handled nets.</td>
</tr>
<tr>
<td>Arbitrary Cycles</td>
<td>-</td>
<td>Not supported. TMS does not allow to go back to previously executed activities. Also, there cannot be more than one entry and exit points.</td>
</tr>
<tr>
<td>Implicit Termination</td>
<td>-</td>
<td>Not supported because processes are block structured with a single start and end node.</td>
</tr>
<tr>
<td>Multiple Instances without Synchronization</td>
<td>-</td>
<td>Not supported. The only way to create multiple instances is by parallel split of sequence activities with simple merge and synchronization.</td>
</tr>
<tr>
<td>Pattern</td>
<td>Rating</td>
<td>Motivation</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Multiple Instances with a Priori Design-Time Knowledge</td>
<td></td>
<td>Not supported. No means of designating that multiple instances of an activity are required.</td>
</tr>
<tr>
<td>Multiple Instances with a Priori Run-Time Knowledge</td>
<td>+</td>
<td>Supported by inserting <em>Job Item Sequence</em> or <em>Work Item Sequence</em> constructs. A new instance of each activities therein will then be executed on per language or per file basis as defined when starting a new job. They will run in parallel and be synchronized at completion. At the core of basic translation workflows.</td>
</tr>
<tr>
<td>Multiple Instances without a Priori Run-Time Knowledge</td>
<td>-</td>
<td>Not supported. It is not possible to create new instances during execution. It is not possible to add new files or new languages to a running job although this may be requirement. Instead, a new job needs to be created.</td>
</tr>
<tr>
<td>Deferred Choice</td>
<td>-</td>
<td>Not supported. Executing branches must run to completion, unless an exception occurs. There is no support for races between different branches.</td>
</tr>
<tr>
<td>Interleaved Parallel Routing</td>
<td>-</td>
<td>Not supported. There is no way to interleave activities without specifying an order.</td>
</tr>
<tr>
<td>Milestone</td>
<td>+</td>
<td>Supported by the pivot activity. The Pivot activity allows to implement processes in which translation to some target languages requires another target language to be translated first (cf. Pivot process in Appendix F)</td>
</tr>
<tr>
<td>Cancel Activity</td>
<td>+</td>
<td>Supported. <em>Stop</em> activity is used for cancelling a <em>Job</em>, <em>JobItems</em> or <em>Work Items</em> in a <em>Job</em> while the workflow is running. Implies that the files in a given locale are definitively removed in the remaining activities of the <em>Job</em>. Affected files are not available for delivery.</td>
</tr>
<tr>
<td>Pattern</td>
<td>Rating</td>
<td>Motivation</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cancel Case</td>
<td>-</td>
<td>Not supported. It is not possible to skip, redo activities or withdraw an activity in one branch triggered by an activity in another branch.</td>
</tr>
<tr>
<td>Structured Loop</td>
<td>-</td>
<td>Not supported. TMS does not allow to go back to previously executed activities.</td>
</tr>
<tr>
<td>Recursion</td>
<td>-</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Transient Trigger</td>
<td>-</td>
<td>Not supported. There is no means of triggering an activity from outside the process instance.</td>
</tr>
<tr>
<td>Persistent Trigger</td>
<td>-</td>
<td>Not supported. There is no means of triggering an activity from outside the process instance. The opposite is possible to a limited though: using the TMS API to perform steps outside of TMS when a Job complete signal is received. Support for this pattern and Cancel Case pattern would allow Control to skip steps in workflow explicitly triggered by a user (cf. chapter section 5.4.2.2).</td>
</tr>
<tr>
<td>Cancel Region</td>
<td>-</td>
<td>Not supported. Only specific steps after a Stop activity can be cancelled as per Cancel Activity pattern.</td>
</tr>
<tr>
<td>Cancel Multiple Instance Activity</td>
<td>-</td>
<td>Not supported. Only specific steps after a Stop activity can be cancelled as per Cancel Activity pattern. This would apply to all instances, not just select ones.</td>
</tr>
<tr>
<td>Complete Multiple Instance Activity</td>
<td>-</td>
<td>Not supported. Cannot forcibly complete activities in any given branch.</td>
</tr>
<tr>
<td>Blocking Discriminator</td>
<td>-</td>
<td>Not supported. No ability to block activity triggering.</td>
</tr>
<tr>
<td>Cancelling Discriminator</td>
<td>-</td>
<td>Not supported. There is no support for the Discriminator pattern or any ability to cancel a set of preceding activities.</td>
</tr>
<tr>
<td>Pattern</td>
<td>Rating</td>
<td>Motivation</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Structured N-out-of-M Join</td>
<td>-</td>
<td>Not supported. Only matching pair of splits/joins are supported, requirement for well-handled nets.</td>
</tr>
<tr>
<td>Blocking N-out-of-M Join</td>
<td>-</td>
<td>Not supported. No ability to block activity triggering.</td>
</tr>
<tr>
<td>Cancelling N-out-of-M Join</td>
<td>-</td>
<td>Not supported. There is no ability to cancel a set of preceding activities.</td>
</tr>
<tr>
<td>Generalised AND-Join</td>
<td>-</td>
<td>Not supported. Only safe Petri net diagrams can be used where the activities in all incoming branches contain only 1 token.</td>
</tr>
<tr>
<td>Static Partial Join for Multiple Instances</td>
<td>-</td>
<td>Not supported. The next activity can only be commenced when all parallel branches have completed due to support limited to only matching pair of splits/joins.</td>
</tr>
<tr>
<td>Cancelling Partial Join for Multiple Instances</td>
<td>-</td>
<td>Not supported. There is no ability to conditionally cancel activities in certain parallel <em>Activity sequence</em> branches.</td>
</tr>
<tr>
<td>Dynamic Partial Join for Multiple Instances</td>
<td>-</td>
<td>Not supported. There is no means of adding further instances to a multi-instance task once started.</td>
</tr>
<tr>
<td>Acyclic Synchronizing Merge</td>
<td>-</td>
<td>Not supported. There is no means to control synchronization as per <em>Synchronization</em> pattern.</td>
</tr>
<tr>
<td>General Synchronizing Merge</td>
<td>-</td>
<td>Not supported. If any exception on an activity in any parallel branches, it must be resolved by explicit user action in the <em>Alerts</em> view so that synchronization of all branches occur and workflow execution at the merge can continue.</td>
</tr>
<tr>
<td>Pattern</td>
<td>Rating</td>
<td>Motivation</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Critical Section</td>
<td>-</td>
<td>Not supported. There is no such communication allowed between parallel branches, the extent of which has been detailed in the <em>Milestone</em> pattern.</td>
</tr>
<tr>
<td>Interleaved Routing</td>
<td>-</td>
<td>Not supported. There is no way to interleave activities without specifying an order as detailed in the <em>Interleaved Parallel Routing</em> pattern.</td>
</tr>
<tr>
<td>Thread Merge</td>
<td>-</td>
<td>Not supported. Multi-thread not supported in a single branch where activities can only be executed sequentially at any given time.</td>
</tr>
<tr>
<td>Thread Split</td>
<td>-</td>
<td>Not supported as per <em>Thread Merge</em> pattern (cf. no custom branching limitation in chapter section 5.4.2.3)</td>
</tr>
<tr>
<td>Explicit Termination</td>
<td>+</td>
<td>Directly supported. Using <em>Cancel Activity</em> pattern, the remaining activities in a given branch are cancelled; if used in a <em>Job Item</em> or <em>Work Item</em> sequence, the job waits for all parallel branches to reach a <em>Stop</em> activity to cancel the entire job.</td>
</tr>
</tbody>
</table>
Appendix H: Exception handling patterns TMS evaluation

The tables below evaluate the support of each exception type. There is no support for *External Triggers* or *Constraints violations* exception types.

**Work Item Failure**

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFF-RAC-COM</td>
<td>A job status shows <em>Start Failed</em> if an issue with underlying workflow configuration. The job is immediately terminated with no possibility of restart.</td>
</tr>
<tr>
<td>SFF-RAC-RBK</td>
<td>A job can be cancelled by the user with relevant privilege at any time before job completion, cancelling all activities in progress or pending execution. Job status then shows <em>Pending cancellation</em> for 24h during which rollback is possible; after that, the job cannot be un-cancelled.</td>
</tr>
<tr>
<td>SFF-RCC-NIL</td>
<td>If a file gets corrupted by a human (i.e. during check in) or by a system activity, an error will be displayed in the <em>Alerts</em> tab of the <em>Status</em> view but only if a later system activity throws an exception whilst handling the file content (cf. Basic support for task errors handling in chapter section 5.4.5.1)</td>
</tr>
<tr>
<td>SFF-RCC-NIL</td>
<td>An error is thrown into the <em>Alerts</em> tab of the <em>Status</em> view if a failure is returned from an external application such as Logoport TM or Translation Workspace. Remaining activities in current branch are suspended until user action the error (skip, restart, retry) changing files at failed step (check-out and check-in) or not.</td>
</tr>
</tbody>
</table>
### Deadline Expiry

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCO-CWC-NIL</td>
<td>A human task status is marked by the system as overdue if it has not been finished and in the allocated schedule options (Duration, Start Threshold Delay or End Threshold Delay) of the associated human activity setting.</td>
</tr>
<tr>
<td>ACA-CWC-NIL</td>
<td></td>
</tr>
<tr>
<td>SCE-CWC-NIL</td>
<td></td>
</tr>
<tr>
<td>ORO-CWC-COM</td>
<td>Fallback options of Planning activities determine who to reassign when delay has been reached (work hours, time value). Behaviour is to set as unclaimed, set as new task, reassign to job contact or leave Assigned status.</td>
</tr>
<tr>
<td>ACA-CWC-NIL</td>
<td>Certain activities connect to external applications. This may cause issues if said application hang and does not return a timeout message. TMS does not seem to support explicit timeout for those activities (cf. No explicit timeout for external applications in chapter section 5.4.5.2) and prevent correct workflow execution.</td>
</tr>
</tbody>
</table>

### Resource Unavailable

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARO-CWC-RBK</td>
<td>Manually unclaimed tasks by translators are displayed on the TMS Planning dashboard. Task can then be claimed by the original user or users of the same pool.</td>
</tr>
<tr>
<td>ARA-CWC-NIL</td>
<td>Tasks are unassigned manually by a user with certain privilege (PM).</td>
</tr>
<tr>
<td>ORO-CWC-COM</td>
<td>In priority mode, if first prioritized assignee has not accepted a task before a certain time (Fallback delay), then task is reassigned to next priority level.</td>
</tr>
<tr>
<td>OCO-CWC-NIL</td>
<td>Tasks still not accepted by anyone are still available for other users to accept based on user assignment options.</td>
</tr>
</tbody>
</table>
Appendix I: New project Engineering template

Once a PM hands-off a localisation kit to the engineer team, a number of information are requested to perform the job such as:

**General project details**

- Source language
- Target languages (locales)
- Source files
- Target files
- Reference material
- TM name, server, folder
- Query management
- Schedule / milestones / deadline
- Deliverables
**Engineering-specific details**

- Detailed instructions
- Tools and software for ENG, version requirements
- Priority list (locales, modules)
- Info about translation scope (e.g. images, videos)
- Format for analysis and translation (RTF, XLZ, etc.)
- Splitting files (If yes, set if split by number of files or words)
  
<table>
<thead>
<tr>
<th>Split by # of:</th>
<th>Files</th>
<th>Words</th>
</tr>
</thead>
</table>

- Assign specific file(s) to a given translator
- Lock frequent segments (If yes, set minimum frequency and word count as below)

<table>
<thead>
<tr>
<th>Minimum Frequency</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Wordcount</td>
<td>3</td>
</tr>
</tbody>
</table>

- Format of word counts log. Many LSP issue different set of logs for the client and their vendors, making money in-between. If Lock Frequent Segment (LFS) is used, PM may want to run the client logs on the files before LFS and the vendor logs on the files after LFS to increase the profit margin.

- Pre-translate
  
  Pre-translate files against the TM. If yes, the PM needs to decide whether or not to lock 100% of 101% matches (context matches) and if only 101% matches are pre-translated (by default 100% and 101% matches are leveraged in the target).

<table>
<thead>
<tr>
<th>Pretranslation Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock 100% Matches</td>
</tr>
<tr>
<td>Lock 101% Matches</td>
</tr>
<tr>
<td>Translate Only 101% Matches</td>
</tr>
</tbody>
</table>

- Market-specific information (e.g. localized URLs, regional images)