Project no. FP6-028038

04//06//2007

Palette

Pedagogically sustained Adaptive LEarning Through the exploitation of Tacit and Explicit knowledge

Integrated Project

Technology-enhanced learning

D.INF.04 – First version of the information reuse service

Due date of deliverable: May 31, 2007
Actual submission date: June 11, 2007

Start date of project: February 01, 2006   Duration: 36 months

Organisation name of lead contractor for this deliverable: EPFL

Project co-funded by the European Commission within the Sixth Framework Programme		
Dissemination Level		
R	Public	**PU**

Keyword List: Prototype, Document reuse, Schema matching, Structure transformations
Responsible partner: Aida Boukottaya, UNIFR

MODIFICATION CONTROL			
Version	Date	Status	Modifications made by
1.0	10/05/2007	Draft	Aida Boukottaya
2.0	20/05/2007	Draft	Aida Boukottaya, Bastien Vaucher
3.0	25/05/2007	Draft	Aida Boukottaya (version sent to evaluators)
4.0	10/06/2007	Final	Aida Boukottaya

Deliverable manager
- Aida Boukottaya, UNIFR

List of Contributors
- Aida Boukottaya, UNIFR
- Bastien Vaucher, EPFL

List of Evaluators
- Bernadette Charlier, UNIFR
- Vincent Quint, INRIA

Summary

This deliverable describes the prototype version of **DocReuse**[1], a web-based application that supports the reuse of structured documents. It also defines the functional specifications of the tool. The specifications include its main functions and features as well as an operational scenario of use that justifies the need of providing such features to communities of practices. We conclude with interesting issues for the future and the planned steps to complete the development of DocReuse.

---

[1] The prototype is in a demonstrable form, and can be tested through some examples (http://docreuse.epfl.ch/release.html).

TABLE OF CONTENT

1. Introduction

One of the fundamental goals we address in the context of WP2 is to enable and entice CoP members to author structured documents. To reach this goal, WP2 partners (INRIA) propose an editing tool (Amaya) that enables CoP members to produce structured documents in an easy manner without dealing with language syntax. CoP members are motivated to structure their documents, especially if they experience services or applications that consume the structured documents and immediately produce tangible and useful results. One of these applications is DocReuse.

DocReuse (Document Reuse) is a tool enabling the semi-automatic reuse of *structured documents*. Document reuse addresses the problem of restructuring existing documents initially structured for a given purpose to be used in a different context, thus improving reusability and information sharing within communities of practice. This is typically attained in real world by writing translators encoded on a case-by-case basis which is time consuming and generally requires programming skills. Our focus is to automate as much as possible this process. A notion closely tied with structured document reuse is that of structure transformations. Schema matching[2] is a critical step in structured document transformations. Manual matching is expensive and error-prone. It is therefore important to develop techniques to automate the matching process and thus the transformation process.

As described in D.INF.01 [1], the initial version of DocReuse takes two XML schemas[3] as input and produces a mapping describing the similarities between those schemas. This output serves as the basis for the automatic generation of transformation scripts. In order to define the appropriate evolution of the initial version of DocReuse in the context of Palette, an extensive review of CoP members' interviews was conducted towards the specification of their needs and requirements in term of information reuse. Based on this analysis, a set of

---

[2] Schema matching is the task of finding correspondences between two heterogeneous schemas (the correspondences are called *mappings*). The mappings enable to deduce automatically the transform scripts which can rearrange and modify the associated data.

[3] http://www.w3.org/XML/Schema

Use-cases involving DocReuse and related to several CoPs (e.g., UX11, Doctoral program Lancaster, Lean-Nett, and Did@cTIC) was produced. Discussions with CoPs representatives and mediators in the framework of the teams A and B have been taken into account in the production of the DocReuse prototype.

One fundamental observation was the difficulty to produce and manipulate XML schemas (which impose a very rigid structure and require being familiar with schema languages). For this, an editing tool Amaya was introduced to CoP members in the context of WP2 in order to help them to describe document structure in an easy manner. This has been done based on the notion of *template*. "A template is a document with some fixed contents and "holes" where the user can insert information. In some cases templates look like a form as those provided by content editors. A template defines the skeleton of the document. It declares components that are specific to the kind of document. By filling the template, the user produces in a transparent manner a class of structured documents (so-called instances) that share the same characteristics." Examples of templates could be found in D.INF.02.

Based on these observations, we essentially focus on the following issues:

- Begin to work with Amaya templates[4] (expressed with XTiger language) as input instead of XML schemas,
- Refine the main services that the DocReuse prototype could offer to answer CoPs needs (see section 2),
- Conduct the implementation based on development guidelines (elaborated in the context of WP5),
- Enhance the user-interface of the DocReuse prototype to facilitate its "acceptability" by CoP members.

This deliverable reports on the first release of the DocReuse tool. Its purpose is to demonstrate the main features of the DocReuse prototype in order to make experiments, to collect users' feedback and to prepare the next release, which will offer a broader support for the above issues.

---

[4] http://www.w3.org/Amaya/Templates/Overview.html

2. General description and main features of DocReuse

The main goal of DocReuse is to enable the automatic reuse of structured documents' content[5]. More precisely, DocReuse aims to offer the following features to CoP members:

- Structure transformation in a transparent manner,

- Producing structured documents through the reuse process,

- Structured visualisation of the produced mapping result enabling the end-user to graphically validate or modify the result,

- Efficient building of a memory (templates/mapping files), which can be reused in future structure transformations,

- Easy access via all major web browsers.

The current architecture of the DocReuse prototype consists of two separate modules: *Template-based restructuring* and *automatic matching and transformation*. The first module offers the possibility to CoP members to restructure their existing HTML resources according to a predefined Amaya template. The second module is composed of several sub-modules (as described in section 2.2). This modular approach enables the easier conversion to a set of *atomic services* implemented as w*eb services* in order to ensure the interoperability with other Palette services and external services (available within CoPs environments). The conversion will be done according to guidelines and standards described in Task1 of WP5. Figure 1 illustrates the global architecture of the DocReuse prototype. Modules are represented as a set of services.

In section 2, we describe DocReuse services as illustrated in Figure 1 and precise the current state of each module in term of implementation progress. The described services address CoP needs in term of information reuse and are deduced directly from the analysis of CoPs synthesis. Section 3 shows how the described services answer CoPs needs through an operational scenario.

---

[5] Structured documents in this deliverable refer to instances produced according to an Amaya template.
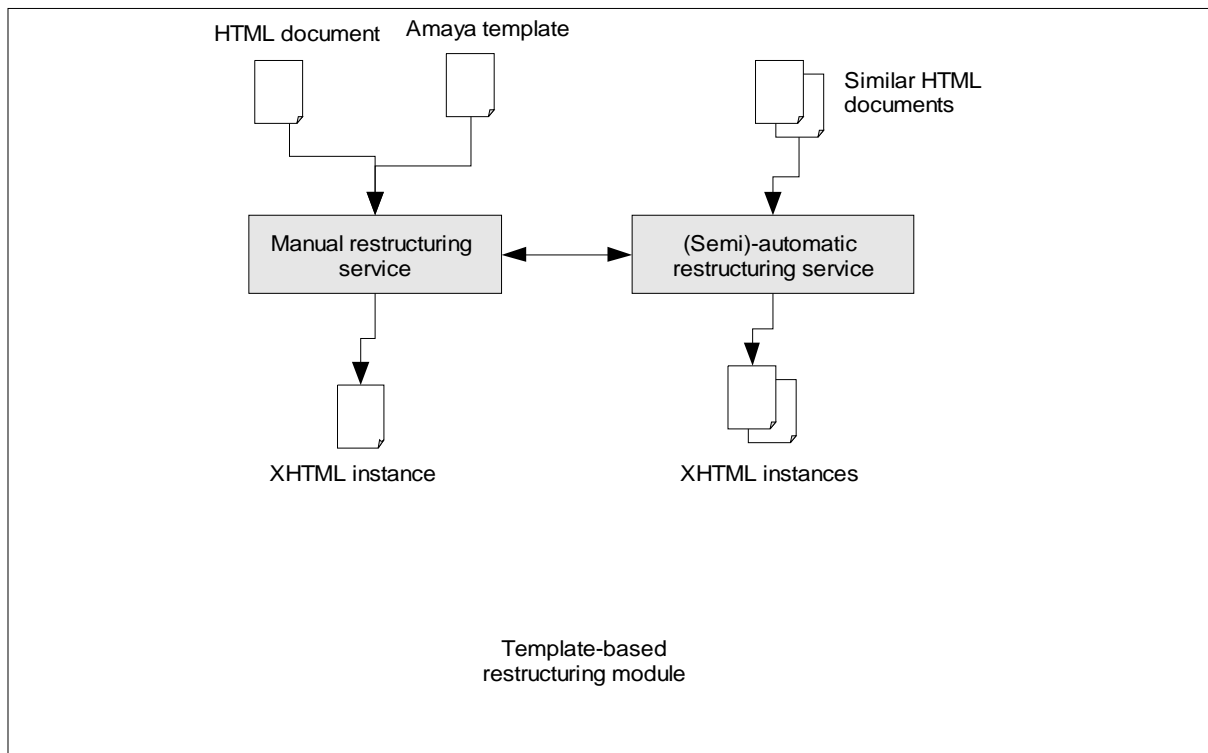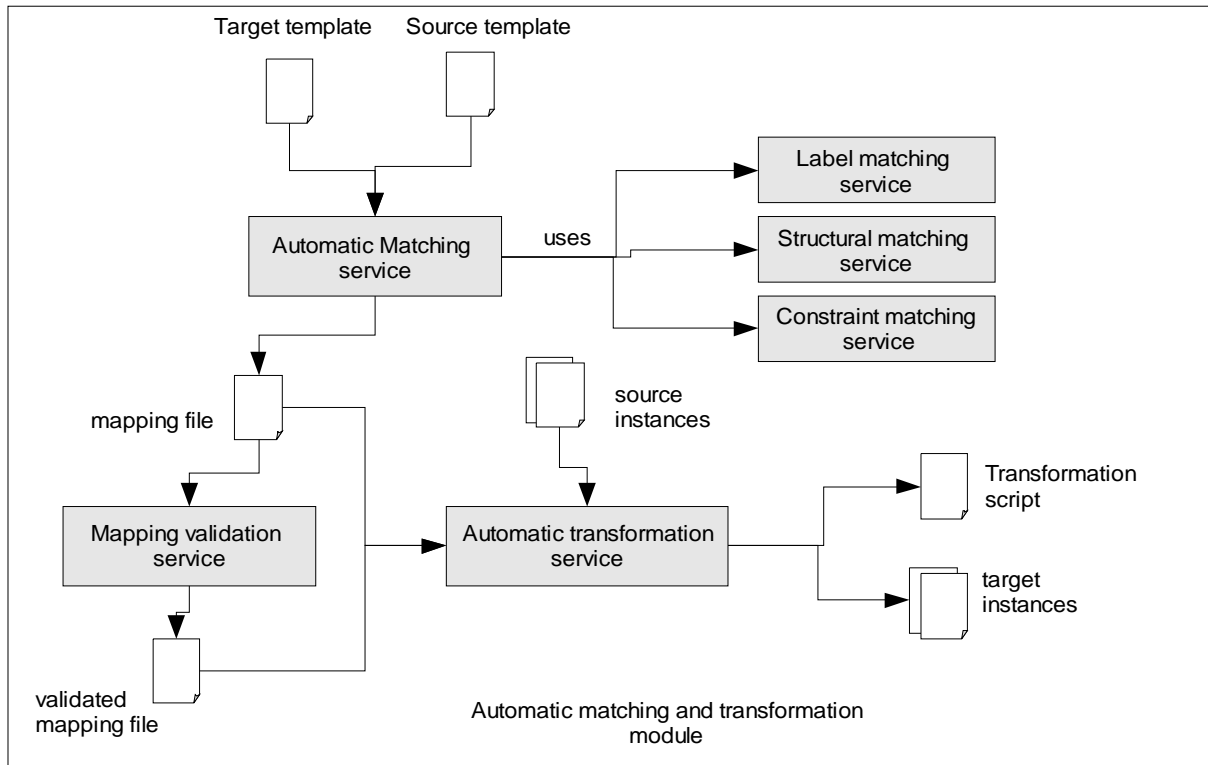
Figure 1: DocReuse services

2.1 Template-based restructuring module:

DocReuse proposes a solution to automatically reuse structured documents. Through CoPs interviews and synthesis (available in the context of Palette[6]), we noticed that although CoP members are motivated to produce structured documents, they also express the need to reuse existing resources in (semi) automatic manner.

Much of the available CoPs' data that we focus on (e.g., contact information, course scheduling, publications, meeting minutes, etc.) already reside in HTML pages, and the challenge is to entice users to make the effort of structuring data. The key idea underlying the template-based restructuring module is to address a scenario where the data comes *before* the schema. In order to entice people to structure their data, we offer to CoP members a set of predefined templates to which they can map their data easily (see figure 1). Once a given document has been structured by the user, similar documents could be then structured automatically. As an example, let us consider a set of meeting minutes within a community of practice. These minutes generally have the same components (author, participants, main decisions, etc,). A user could restructure one meeting minutes interactively then all the existing meetings minutes could be restructured automatically.

2.1.1 Implementation state:

As stated earlier, the template-based restructuring module involves two services: *the manual restructuring service* and *the (semi)-automatic restructuring service* (see figure 1). Currently, only the manual restructuring service is implemented[7]. Given a HTML file and a template, the user is able to manually drag and drop content from the HTML file to instantiate the template. The template appears as a form that defines the skeleton of the document. Once the template

---

[6] https://bscw.ercim.org/bscw/bscw.cgi/172617

[7] We provide a very first prototype for the manual restructuring service. The goal is to study the need of such service and CoPs requirements. The implementation (in term of features and user-interface) is not yet complete. We will rely on CoPs request and feedback to make future decisions about the service features.

is filled, a structured instance (XHTML instance) is produced. The produced instance follows the structure described in the template.

Moreover, the system allows the user to build a memory concerning his restructuring activities. In fact, the user can create several projects. Each project includes a set of templates, restructured instances and instances to be restructured. Several implementation improvements are planned:

- At the moment only the structural organisation of the restructured instance is validated according to the chosen template. Datatypes validation will be implemented in the future according to XTiger Datatypes.
- Currently the user should complete the restructuring activity in one step. In the future, we aim to enabling CoP members to perform this activity in several steps which facilitate the restructuring of huge HTML documents.
- Enhance the user-interface based on CoPs feedback.

2.1.2 Example of use:

The following table summarises the different steps a CoP member should go through in order to restructure a set of HTML documents according to a given template.

The home page offers to the CoP member a global view of the latest projects he is involved in.

To begin a restructuring activity, he has to click on the link "Restructuring".

	After that, the user selects the unstructured HTML document and the desired template.
	Now, the user sees the HTML document on the left and the template on the right. For the restructuring, he just has to select the text in the HTML document and to drag it in the desired text area of the template.
	At the end, the user can see a preview of the restructured document and definitively save the document by clicking on the button "Save".

2.2 Automatic matching and transformation module:

This module takes as input two Amaya templates (source and target), a set of source instances and produces automatically structured target instances (see figure 1). The goal is to enable

CoP members to transform structured documents in a transparent and automatic manner. The automatic transformation is based on a mapping file produced automatically by the system.

This module involves three services:

Automatic matching service: this service is the core of the DocReuse tool. The goal of this service is to identify the parts to be transformed in the reusable documents and to describe how to place their content in appropriate context within the target template. The result is a set of mappings (correspondences) relating facts from the source documents with the target template by encapsulating all necessary information to transform source documents into instances of the target template. In order to achieve this goal, we adopt a multi-criteria matching process. Each criterion will be represented by a Web service. These services are extensible. As new criterion becomes available, a new service is created. Examples of services include:

- *Label matching:* measures the similarities between entities based on the meaning inferred from their names (e.g., author and writer are synonyms). This service will be enhanced using CoPs specific vocabularies (thesauri), local dictionaries (that define the common set of abbreviations, acronyms, and commonly used substring/short hand notations for a given CoP) as well as CoPs ontologies provided by WP3.
- *Constraint matching:* relates entities based on their respective constraints. Such constraints include the use of Datatypes, occurrence constraints, etc.…
- *Structural matching:* relates entities based on the similarity of the structural context in which they appear. For this templates are modelled as trees (treated as a collection of paths). This has two advantages: (1) a match between two paths can be defined as long as there are some matching nodes along the paths; and (2) a match between two nodes can be defined irrespective of where they occur in the tree as long as there is at least a partial match between their paths.

We have designed an algorithm to combine all the above criteria and produce a mapping result[8] that clearly defines source and target mapped entities, required transformation operations, and conditions under which the mapping can be executed. An example of

---

[8] The detailed algorithm can be found in [2].

mapping file could be found in the appendices. Currently, we only perform a simple label matching algorithm (in the future, we need to take into account semantics described within CoPs ontologies and local dictionaries). The constraint matching is not yet applied to Amaya templates. The structured matching is the core of the system and its implementation is finalised and tested.

Mapping validation service: Experience suggests that fully automated schema matching is infeasible, especially for complex matches that involve transformation operations. Schema matching algorithms must interact with the user in order to produce correct mappings. One of the most important open problems for schema matching is efficient user interaction. Specific user input could be for example interactively requested at critical points where it is maximally useful. Moreover, even if a near perfect matching solution exists, the user still has to validate the produced mapping results. The key is then to discover how to minimize user interaction but maximizing the impact of the feedback. For this, the validation will be done in a graphical manner. A theoretical study about user interaction is described in [4]. We need to adapt this study to the current prototype taking into account CoP members needs in term of validation. This service is not yet implemented and will be one of our focuses in next releases.

Automatic transformation service:

This service generates automatically the appropriate transformation scripts (XSLT programs). We first have defined a model for structuring mapping results according to a mapping schema that describes the five dimensions of a mapping result: entity, cardinality, structural, transformation and constraint dimensions. Second we have designed an algorithm that generates automatically XSLT scripts based on the above mapping structure. For each matching node pair, the algorithm traverses the target schema graph in a depth-first manner and generates progressively transformation rules. The service is already implemented. Templates, produced mapping results as well as the transformation scripts are stored. When similar reusable documents are available, the transformation process is done automatically without re-applying the entire matching process.

2.3 Summary and functional specifications of DocReuse services:

The following table summarises the functional specifications of DocReuse. For each service we give a brief description, required input and output, possible dependencies on other services (DocReuse services or external Palette services), and implementation state and future work.

DocReuse Services	Description	Input	Output	Dependencies on other services	Implementation state and future work
Manual restructuring service	CoP members can interactively restructure their existing HTML documents according to an Amaya template.	HTML instances, Amaya Template	Restructured instances (XHTML)	Must interpret Amaya templates	A first version is implemented. To be done: Datatypes validation, conversion to Web service.
(Semi)-automatic restructuring service	Based on manual restructuring results, this service restructures similar HTML documents	HTML documents	Restructured Instances (XHTML)	Manual restructuring service	Not yet implemented
Automatic matching service	Automatic generation of a mapping file through the composition of three services (label, constraint and structural matching).	Two Amaya Templates	Mapping file	-Must interpret Amaya templates -Composition of three services (label, constraint and structural matching) -Must interpret and use CoPs ontologies, dictionaries	Partially implemented: -Label matching: not yet implemented. -Constraint matching: not yet implemented -Structural matching: finalised.
Mapping validation service	Graphically validate (modify) the mapping results	Mapping file	Validated mapping file	No	Not yet implemented
Automatic transformation service	Automatic transformation generation.	Mapping file, Source instances	Transformation script, target instances	No	Finalised

3. CoPs needs and Operational scenario:

The goal of this section is to analyse CoPs needs in term of information reuse and to show how DocReuse features could answer the needs. An operational scenario[9] is described.

3.1 Examples of CoPs needs:

The following table gives some examples of CoPs needs in term of information reuse. Based on our observations and discussions with CoPs mediators and representatives, we notice that in order to reuse information, a CoP member (teacher, tutor, learner, etc…) generally goes through a manual process:

1. consult all the available resources (discussions on the forum in the case of UX11, reported problems in the case of Lancaster, tutors discussions in the case of Learn-Nett, and learners journals or groups reflections about practices in the case of Did@ctic);
2. classify them by topic/issue, choose the most relevant ones to be reused,
3. "cut&paste" the reusable information, and
4. adapt it to new needs (e.g,. on the FAQ the questions/answers are classified by topic; no more then three questions/answers per topic; no more then two answers per question, etc.). Moreover, these discussions could be presented according to new layout constraints (for example using a table or a list).

This process is generally time consuming and error-prone. (e.g., a teacher could miss relevant discussions on the forum or make a mistake in classifying questions/answers. Moreover, the teacher will repeat the same process each year). The fourth point (adaptation effort) requires a non negligible effort (the writing of a structure transformation script in the case of structured documents). In the next section, we will show how the process becomes automatic using DocReuse services.

---

[9] Another example of use related to Learn-Nett community could be found in [3]

CoP needs (in term of reuse) (Problem/wishes)
UX11 (*Reusing discussions on forums in order to produce structured FAQs and Hints to support students in their activities***)**
Discussions on the forum could be reused in order to produce FAQs (Frequently Asked Questions) and hints for practical trainings for next year (e.g,. an answer posted on the forum is helpful for a given practical training and teacher decides to incorporate it as a hint for next year).
Lancaster Doctoral Program (*Reorganization reported problems in order to produce reusable "support" for students***)**
Within this community, students can report a problem using a specific form (http://domino.lancs.ac.uk/home.nsf/Problem?OpenForm). These reported problems (essentially technical ones: access, login, passwords...) are used to generate a support document for students (http://domino.lancs.ac.uk/home.nsf/H?OpenView), where a problem is described using a topic, and an answer. In order to keep track of these encountered problems and reuse them to provide more assistance to students, the "reported problems archives" should be reorganized in a better structured way (for example in the form of FAQs or a set of support pages). These FAQs and/or support pages could then be reused for example by students that didn't participate to the discussions or by students of next years. Given the number of reported problems, it's important to automate the process.
Learn-Nett (*Keeping track of discussions and reuse them for providing new materials for tutors***)**
Discussions between tutors reflect their experience and expertise. Capitalisation and reuse of such discussions is of major concern within the Learn-Nett community. For instance, a tutor could ask a question about a specific issue/problem that he encountered, other tutors could give him some advices, reflections, methodologies to follow, etc,. These reflections/hints could then be reused and incorporated either to update the pedagogical guide for tutors, to produce a new guide for next year, or to produce a kind of help manual for tutors. Moreover, students' research papers and other documents could be reused for the design of tutors' tools and for the work of the coordination team.
Did@ctic (*Reuse of learners journals to carry on a meta-journal of the community***)**
The Did@ctic community aims to constitute a set of resources, a series of examples, "vignettes" that can be used and reused to support reflections on teaching or to organise training sessions. Group of learners are formed to conduct reflection about their practices. These discussions are supplemented by an exercise of reflexivity: each participant writes his own "journal", where he can express his personal representations of the life of the community and the learning outcomes. Reusing these reflections to carry on "meta-journal" of the community is very important. The "meta-journal" is a kind of structured synthesis, joining together the various contributions of learners and restructuring them in a coherent and comprehensive manner.

3.2 Operational scenario:

Let us consider the following scenario: a tutor (called tutor 1) (CoP member) classifies the publication activities of his students according to the template 2 (see in the appendices), which requires publications to be clustered by authors. Another tutor (called tutor 2) performs the same activity using a different template (template 1 in the appendices). Tutor 2 wants to reuse the bibliographic data of tutor 1 in order to give more references and support to his students. In order to reuse these data, the user goes generally through a long process. First, he has to understand template 2 structure and semantics. Second, he has to discover correspondences between template 2 and template 1. Such correspondences, also called mappings, identify what data can be useful for him. Examples of such mappings are: both files have an "author" element. Author "name" can be obtained by the concatenation of elements "FirstName" and "LastName". Element "Article" corresponds to element "Journal-Article", etc. Finally, tutor 2 has to choose the transformation language, and to write a set of adequate operations to translate the discovered mapping into a transformation script. Tutor 2 has to generate a list of "author" elements, each containing a sub list of their publications elements "book" and "articles". The process is time consuming, error-prone and requires programming skills.

The following table summarises the process that tutor 2 goes through using the DocReuse tool[10]. He just needs to upload the two templates (template 1 and template 2). The matching module will deduce automatically a mapping file describing the correspondences between the two templates. Based on this result, a transformation script is automatically generated.

---

[10] In this DocReuse release, we provide two examples through predefined Amaya templates (http://docreuse.epfl.ch/demos). The goal is to test the matching and structure transformations services.

DocReuse *Fresh Paint* **Projects list** You have no project for the moment **New Project** Name : [New project] [Create]	Create a new project, which will contain all files related to a specific transformation (template1, template2, instances, mapping file, XSLT file)
New project < Back **1. Templates** Generate mapping Source Target 📄 previous_library.xtd change ? [_____] [Parcourir...] [Upload]	Upload 2 template files. These two templates will be used to generate the mapping file.

DocReuse *Fresh Paint*
 	
`<?xml version="1.0" encoding="UTF-8"?>`
`<MappingResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
`xsi:noNamespaceSchemaLocation=".\mappingSchema.xsd">`
`<HasMappings type="OneToOneMapping" ID="map1"/>`
`<MappingElement>`
`<OneToOneMapping ID="map1">`
`<SourceNode name="/*"/>`
`<TargetNode name="/*"/>`
`<Transformation>`
`<Operation name="connect"/>`
`</Transformation>`
`<HasMappings type="OneToOneMapping" ID="map1.1"/>`
`</OneToOneMapping>`
`</MappingElement>`
`<MappingElement>`
 	
Back to project
Back to home
	Generate mapping and XSLT files. These files are stored on the server, this actually means that later on tutor2 can upload new instances and generate files from without having to regenerate mapping.

2. Instances Generate target instances Sources Targets [_____] [Parcourir...] [Upload]	Upload instances. Instances that will then be converted into the new file structure defined by the mapping/XSLT files.
2. Instances Generate instances **Generate** target instances Sources Targets 🔵 library_instance.html remove	Generate final files using XSLT file. Just by clicking on the "generate target instances", new converted documents are created.

4. Future work:

This deliverable describes DocReuse services based on the observation of Palette CoPs. The presented prototype is the first release of the DocReuse tool. The goal is to collect feedback from Palette partners as well as CoP members. In the future, several issues have to be addressed:

- More support of Amaya templates: testing more templates and working collaboratively with the Amaya team to support all the characteristics of the XTiger language (results will be presented in D.INF.06: "Template-driven transformations (M26)"),

- Refine DocReuse features based on participatory activities with CoPs: this includes namely the analysis of the results produced in D.IMP.03 (a general synthesis of CoPs needs will be done including the questions asked by mediators about Palette tools features),

- Complete the missing implementations (e.g., mapping visualisation, constraint matching, etc…). A special focus will be given to WP5 guidelines (a revised version of those guidelines will be available on M20),

- Focus on enhancing the user interface based on usability studies done in the context of WP1 and WP5.

References

[1] A. Boukottaya, F. Campoy Flores, R. Deltour, V. Quint, Ch. Vanoirbeek, I. Vatton, K. Zouba, D.INF.01: Report on the design of extension mechanisms for creating templates, using templates for editing and customizing the user interface, and of extensions to be integrated in the information reuse tool (M3), 15 May 2006 (pdf, html).

[2] A. Boukottaya, C. Vanoirbeek: Schema matching for transforming structured documents. ACM Symposium on Document Engineering 2005: 101-110.

[3] A. Boukottaya, B. Charlier, C. Vanoirbeek**:** A Document Reuse Tool for Communities of Practice. TEL-CoPs'06, Crete, Greece, October 2, 2006.

[4] A. Boukottaya, Schema matching for structured document transformations. Thèse EPFL, no 3108 (2004).

Appendices

Template 1

```
<?xml version="1.0" encoding="UTF-8"?>
<html     lang="en"     xmlns="http://www.w3.org/1999/xhtml"     xmlns:xt="http://wam.inrialpes.fr/xtiger"
xml:lang="en">
<head>
 <title>University</title>
 <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
 <meta name="description" content="university librairy"/>
 <meta name="keywords" content="librairy university"/>
 <meta name="template_class" content="amaya-cv-en-0.1"/>
   <style type="text/css">
                *{
                        border:0px;
                        padding:0px;
                }
                        h1{
                        display:none;
                }
                        h2{
                        margin-top:0px;
                        margin-bottom:5px;
                        padding:5px;
                }
                        .name h2{
                        background-color:#E0824F;
                        color:#FFF;
                }
                        .address h2{
                        display:none;
                }
                        .author{
                        margin-left:30px;
                        margin-right:30px;
                        clear:both;
                }
```

```
                .library{
                        border:1px solid #E0590F;
                }
                .library h2{
                        background-color:#E0A17E;
                }
                .publications{
                        margin:10px;
                }
                .book_cover{
                        width:50px;
                        height:70px;
                        float:left;
                        border:1px dashed black;
                        margin:0px 5px 5px 5px;
                        text-align:center;
                }
                .end{
                        clear:both;
                }
        </style>
 </head>
<body>
        <div class="university">
                <h1>University</h1>
                <xt:component name="uni">
                        <div class="name">
                                <h2><xt:use types="string" label="name">university_name</xt:use></h2>
                        </div>
                        <div class="address">
                                <h2>Address</h2>
                                <xt:use types="string" label="address">address</xt:use>
                        </div>
                        <div class="library">
                                <h2>Articles</h2>
```

```
<xt:repeat label="articles">
        <xt:component name="article">
        <div class="book_cover">
        <i>cover</i>
        </div>
        <div class="article">
        <div class="title">
        Title : <xt:use types="string" label="title">title</xt:use>
        </div>
        <div class="author">
        <h3>Author</h3>
        <div class="name">
        Name : <xt:use types="string" label="name">name</xt:use>
        </div>
        <div class="address">
        <div class="city">
        City : <xt:use types="string" label="city">city</xt:use>
        </div>
        <div class="state">
        State : <xt:use types="string" label="state">state</xt:use>
        </div>
        <div class="zip">
        Zip : <xt:use types="number" label="zip">0000</xt:use>
        </div>
        </div>
        </div>
        </div>
        </xt:component>
</xt:repeat>
<h2>Journals</h2>
<xt:repeat label="journals">

        <xt:component name="journal">
        <div class="book_cover">
                <i>cover</i>
        </div>
```

```
<div class="journal">
<div class="name">
Name : <xt:use types="string" label="name">journal name</xt:use>
</div>
<div class="editor">
Editor : <xt:use types="string" label="editor">editor name</xt:use>
</div>
</div>
<div class="end"></div>
</xt:component>
</xt:repeat>
<h2>Books</h2>
<xt:repeat label="books">
	<xt:component name="book">
	<div class="book_cover">
		<i>cover</i>
	</div>
	<div class="book">
	<div class="title">
	Title : <xt:use types="string" label="title">title</xt:use>
	</div>
	<div class="price">
	Price : <xt:use types="number" label="price">00.00</xt:use> Frs
	</div>
	<div class="publisher">
	Publisher :<xt:use types="string" label="publisher">publi</xt:use>
	</div>
	<div class="author">
	<h3>Author</h3>
	<div class="name">
	Name : <xt:use types="string" label="name">name</xt:use>
	</div>
	<div class="address">
	<div class="city">
	City : <xt:use types="string" label="city">city</xt:use>
	</div>
```

```
<div class="state">
State : <xt:use types="string" label="state">state</xt:use>
</div>
<div class="zip">
Zip : <xt:use types="number" label="zip">0000</xt:use>
</div>
</div>
</div>
</div>
</xt:component>
</xt:repeat>
</div>
</xt:component>
</div>
</body>
</html>
```

Template 2

```
<?xml version="1.0" encoding="UTF-8"?>
<html lang="en"  xmlns:xt="http://wam.inrialpes.fr/xtiger" xml:lang="en">
<head>
 <title>University</title>
 <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
 <meta name="description" content="University librairy"/>
 <meta name="keywords" content="university librairy"/>
 <meta name="template_id" content="amaya-cv-en-0.1"/>

 <style type="text/css">
          *{
                  border:0px;
                  padding:0px;
          }
          h1{
          margin:0px 0px 10px 0px;
          background-color:#592427;
```

```
                    color:#FFF;
                    padding:2px;
         }
                    .authors{
                    border:1px solid #BB3238;
         }
                     .authors h2{
                    margin:0px 0px 10px 0px;
                    padding:2px;
                    background-color:#BB8A8D;
         }
                    .publications{
                    margin:10px;
         }
        </style>
  </head>
<body>
        <div class="university">
                <xt:component name="uni">
                        <div class="name">
                                <h1><xt:use types="string" label="name">university_name</xt:use></h1>
                        </div>
                        <div class="address">
                                <xt:component name="address">
                                        <div class="city">
                                                City : <xt:use types="string" label="city">city</xt:use>
                                        </div>
                                        <div class="state">
                                                State : <xt:use types="string" label="state">state</xt:use>
                                        </div>
                                        <div class="zip">
                                                Zip : <xt:use types="number" label="zip">0000</xt:use>
                                        </div>
                                </xt:component>
                        </div>
                        <div class="authors">
```

```
<xt:repeat label="authors">
        <xt:component name="author">
        <h2>Author</h2>
        <div class="first_name">
        Firstname:<xt:use types="string" label="firstname">fname</xt:use>
        </div>
        <div class="last_name">
        Last name :<xt:use types="string" label="lastname">lname</xt:use>
        </div>
        <div class="address">
        Address :<xt:use types="string" label="address">address</xt:use>
        </div>
        <div class="publication">
        <h2>Publications</h2>
        <h3>Newspapers-Articles</h3>
        <xt:repeat label="articles">
        <xt:component name="journal-article">
        <div class="abstract">
        Abstract :<xt:use types="string" label="abstract">abstract</xt:use>
        </div>
        <div class="journal">
        Journal :<xt:use types="string" label="journal">journal</xt:use>
        </div>
        <div class="editor">
        Editor :<xt:use types="string" label="editor">editor</xt:use>
        </div>
        </xt:component>
        </xt:repeat>
        <h3>Books</h3>
        <xt:repeat label="books">
        <xt:component name="book">
        <div class="price">
        Price :<xt:use types="string" label="price">price</xt:use>
        </div>
        <div class="title">
        Title :<xt:use types="string" label="title">title</xt:use>
```

```
                                    </div>
                                    <div class="publisher">
                                    Publisher :<xt:use types="string" label="publisher">publi</xt:use>
                                    </div>
                                    </xt:component>
                                    </xt:repeat>
                                    </div>
                                    </xt:component>
                        </xt:repeat>
                </div>
        </xt:component>
    </div>
</body>
</html>
```

Mapping file

```
<?xml version="1.0" encoding="UTF-8"?>

<MappingResult                              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=".\mappingSchema.xsd">


        <HasMappings type="OneToOneMapping" ID="map1"/>


        <MappingElement>
                <OneToOneMapping ID="map1">
                        <SourceNode name="/*"/>
                        <TargetNode name="/*"/>
                        <Transformation>
                                <Operation name="connect"/>
                        </Transformation>
                        <HasMappings type="OneToOneMapping" ID="map1.1"/>
                </OneToOneMapping>
        </MappingElement>


        <MappingElement>
```

```xml
<OneToOneMapping ID="map1.1">
        <SourceNode name="/*/*/div[@class='university']"/>
        <TargetNode name="/*/*/div[@class='university']"/>
        <Transformation>
                <Operation name="connect"/>
        </Transformation>
        <HasMappings type="OneToOneMapping" ID="map1.1.1"/>
    </OneToOneMapping>
</MappingElement>


<MappingElement>
    <OneToOneMapping ID="map1.1.1">
        <SourceNode name="/*/*/div[@class='university']/xt:use[@label='uni']"/>
        <TargetNode name="/*/*/div[@class='university']/xt:use[@label='uni']"/>
        <Transformation>
                <Operation name="connect"/>
        </Transformation>
        <HasMappings type="OneToOneMapping" ID="map1.1.1.1"/>
        <HasMappings type="ManyToOneMapping" ID="map1.1.1.2"/>
        <HasMappings type="NullMapping" ID="map1.1.1.3"/>
    </OneToOneMapping>
</MappingElement>


<MappingElement>
    <OneToOneMapping ID="map1.1.1.1">
        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/*/xt:use[@label='name']"/>
        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/*/xt:use[@label='name']"/>
        <Transformation>
                <Operation name="connect"/>
        </Transformation>
    </OneToOneMapping>
</MappingElement>


<MappingElement>
    <ManyToOneMapping ID="map1.1.1.2">
```

```
                                    <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/div[@class='author']/div[@class='address']"/>
                                    <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='articles']/*/div
[@class='article']/div[@class='author']/div[@class='address']"/>
                                    <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:use[@label='address']"/>
                                <Transformation>
                                        <Operation name="union"/>
                                </Transformation>
                                <Conditions>
                                        <Condition name="Distinct"/>
                                </Conditions>
                                <HasMappings type="ManyToOneMapping" ID="map1.1.1.2.1"/>
                                <HasMappings type="ManyToOneMapping" ID="map1.1.1.2.2"/>
                                <HasMappings type="ManyToOneMapping" ID="map1.1.1.2.3"/>
                        </ManyToOneMapping>
                </MappingElement>


                <MappingElement>
                        <ManyToOneMapping ID="map1.1.1.2.1">
                                    <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/div[@class='author']/div[@class='address']/*/xt:use[@label='city']"/>
                                    <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='articles']/*/div
[@class='article']/div[@class='author']/div[@class='address']/*/xt:use[@label='city']"/>
                                    <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:use[@label='address']/*/xt:use[@label='city']"/>
                                <Transformation>
                                        <Operation name="merge"/>
                                </Transformation>
                        </ManyToOneMapping>
                </MappingElement>


                <MappingElement>
                        <ManyToOneMapping ID="map1.1.1.2.2">
```

```
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/div[@class='author']/div[@class='address']/*/xt:use[@label='state']"/>
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='articles']/*/div
[@class='article']/div[@class='author']/div[@class='address']/*/xt:use[@label='state']"/>
                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:use[@label='address']/*/xt:use[@label='state']"/>
                            <Transformation>
                                <Operation name="merge"/>
                            </Transformation>
                    </ManyToOneMapping>
            </MappingElement>


            <MappingElement>
                    <ManyToOneMapping ID="map1.1.1.2.3">
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/div[@class='author']/div[@class='address']/*/xt:use[@label='zip']"/>
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='articles']/*/div
[@class='article']/div[@class='author']/div[@class='address']/*/xt:use[@label='zip']"/>
                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:use[@label='address']/*/xt:use[@label='zip']"/>
                            <Transformation>
                                <Operation name="merge"/>
                            </Transformation>
                    </ManyToOneMapping>
            </MappingElement>


            <MappingElement>
                    <NullMapping ID="map1.1.1.3">
                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']"/>
                            <HasMappings type="ManyToOneMapping" ID="map1.1.1.3.1"/>
                    </NullMapping>
            </MappingElement>
            <MappingElement>
                    <ManyToOneMapping ID="map1.1.1.3.1">
```

```
                              <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/div[@class='author']"/>
                              <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='articles']/*/div
[@class='article']/div[@class='author']"/>
                              <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']"
/>

                           <Transformation>

                                 <Operation name="union"/>

                           </Transformation>

                           <Conditions>

                                 <Condition name="Distinct"/>

                           </Conditions>

                           <HasMappings type="NullMapping" ID="map1.1.1.3.1.1"/>

                           <HasMappings type="OneToManyMapping" ID="map1.1.1.3.1.2"/>

                           <HasMappings type="NullMapping" ID="map1.1.1.3.1.4"/>

                     </ManyToOneMapping>

               </MappingElement>

               <MappingElement>

                     <NullMapping ID="map1.1.1.3.1.1">

                           <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']"/>

                                 <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.1"/>

                                 <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.2"/>

                     </NullMapping>

               </MappingElement>

               <MappingElement>

                     <OneToOneMapping ID="map1.1.1.3.1.1.1">

                           <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='articles']"/>

                           <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='articles']"/>

                           <Transformation>

                                 <Operation name="connect"/>

                           </Transformation>

                           <Conditions>
```

```
                                        <Condition    value="html/university/uni/library/articles  [author  =  current
articles ]"/>

                        </Conditions>

                        <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.1.1"/>

                </OneToOneMapping>

        </MappingElement>


        <MappingElement>

                <OneToOneMapping ID="map1.1.1.3.1.1.1.1">

                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='journals']/*/di
v[@class='journal']"/>

                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='articles']/xt:use[@label='journal-article']"/>

                        <Transformation>

                                <Operation name="connect"/>

                        </Transformation>

                        <HasMappings type="NullMapping" ID="map1.1.1.3.1.1.1.1.1"/>

                        <HasMappings type="NullMapping" ID="map1.1.1.3.1.1.1.1.2"/>

                        <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.1.1.3"/>

                </OneToOneMapping>

        </MappingElement>


        <MappingElement>

                <NullMapping ID="map1.1.1.3.1.1.1.1.1">

                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='articles']/xt:use[@label='journal-
article']/*/xt:use[@label='abstract']"/>

                </NullMapping>

        </MappingElement>


        <MappingElement>

                <NullMapping ID="map1.1.1.3.1.1.1.1.2">

                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='articles']/xt:use[@label='journal-
article']/*/xt:use[@label='journal']"/>

                </NullMapping>
```

```
        </MappingElement>


        <MappingElement>
                <OneToOneMapping ID="map1.1.1.3.1.1.1.1.3">
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='journals']/*/di
v[@class='journal']/*/xt:use[@label='editor']"/>
                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='articles']/xt:use[@label='journal-
article']/*/xt:use[@label='editor']"/>
                        <Transformation>
                                <Operation name="connect"/>
                        </Transformation>
                </OneToOneMapping>
        </MappingElement>


        <MappingElement>
                <OneToOneMapping ID="map1.1.1.3.1.1.2">
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']"/>
                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='books']"/>
                        <Transformation>
                                <Operation name="connect"/>
                        </Transformation>
                        <Conditions>
                                <Condition value="html/university/uni/library/books [author = current books
]"/>
                        </Conditions>
                        <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.2.1"/>
                </OneToOneMapping>
        </MappingElement>


        <MappingElement>
                <OneToOneMapping ID="map1.1.1.3.1.1.2.1">
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']"/>
```

```
                    <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='books']/xt:use[@label='book']"/>
                        <Transformation>
                            <Operation name="connect"/>
                        </Transformation>
                        <Conditions>
                            <Condition value="html/university/uni/library/books/book [author = current
book ]"/>
                        </Conditions>
                        <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.2.1.1"/>
                        <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.2.1.2"/>
                        <HasMappings type="OneToOneMapping" ID="map1.1.1.3.1.1.2.1.3"/>
                    </OneToOneMapping>
            </MappingElement>


            <MappingElement>
                    <OneToOneMapping ID="map1.1.1.3.1.1.2.1.1">
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/*/xt:use[@label='price']"/>
                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='books']/xt:use[@label='book']/*/xt:use[@label='price']"/>
                        <Transformation>
                            <Operation name="connect"/>
                        </Transformation>
                    </OneToOneMapping>
            </MappingElement>


            <MappingElement>
                    <OneToOneMapping ID="map1.1.1.3.1.1.2.1.2">
                        <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/*/xt:use[@label='title']"/>
                        <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='books']/xt:use[@label='book']/*/xt:use[@label='title']"/>
                        <Transformation>
                            <Operation name="connect"/>
```

```
            </Transformation>
        </OneToOneMapping>
    </MappingElement>
    <MappingElement>
        <OneToOneMapping ID="map1.1.1.3.1.1.2.1.3">
            <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/*/xt:use[@label='publisher']"/>
            <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
div[@class='publication']/xt:repeat[@label='books']/xt:use[@label='book']/*/xt:use[@label='publisher']"/>
            <Transformation>
                <Operation name="connect"/>
            </Transformation>
        </OneToOneMapping>
    </MappingElement>
    <MappingElement>
        <OneToManyMapping ID="map1.1.1.3.1.2">
            <SourceNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/div[@class='library']/xt:repeat[@label='books']/*/div[
@class='book']/div[@class='author']/*/xt:use[@label='name']"/>
            <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
*/xt:use[@label='first_name']"/>
            <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
*/xt:use[@label='last_name']"/>
            <Transformation>
                <Operation name="split" param="WS"/>
            </Transformation>
        </OneToManyMapping>
    </MappingElement>
    <MappingElement>
        <NullMapping ID="map1.1.1.3.1.4">
            <TargetNode
name="/*/*/div[@class='university']/xt:use[@label='uni']/*/xt:repeat[@label='authors']/xt:use[@label='author']/
*/xt:use[@label='address']"/>
        </NullMapping>
    </MappingElement>
</MappingResult>
```