<table>
<thead>
<tr>
<th>Project Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Hashtag</strong></td>
</tr>
<tr>
<td><strong>Project Title (and acronym)</strong></td>
</tr>
<tr>
<td><strong>Start Date</strong></td>
</tr>
<tr>
<td><strong>End Date</strong></td>
</tr>
<tr>
<td><strong>Lead Institution</strong></td>
</tr>
<tr>
<td><strong>Project Director</strong></td>
</tr>
<tr>
<td><strong>Project Manager &amp; contact details</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Partner Institutions</strong></td>
</tr>
<tr>
<td><strong>Project Web URL</strong></td>
</tr>
<tr>
<td><strong>Programme Name (and number)</strong></td>
</tr>
<tr>
<td><strong>Programme Manager</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
</tr>
<tr>
<td><strong>Project Role(s)</strong></td>
</tr>
<tr>
<td><strong>Date</strong></td>
</tr>
<tr>
<td><strong>Filename</strong></td>
</tr>
<tr>
<td><strong>URL</strong></td>
</tr>
<tr>
<td><strong>Access</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document History</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
</tr>
</tbody>
</table>
Table of Contents

1) ACKNOWLEDGEMENTS ........................................................................................................ 3

2) REPORT SUMMARY ......................................................................................................... 4
   2.1 PROJECT OVERVIEW .................................................................................................... 4
   2.2 PROJECT OUTPUTS ....................................................................................................... 5
   2.3 IMPACT AND BENEFITS TO THE COMMUNITY ...................................................... 7
   2.4 MAIN LESSONS LEARNT............................................................................................. 8

3) MAIN BODY OF REPORT ................................................................................................ 9
   3.1 WHAT DID YOU DO? (METHODOLOGY) ................................................................... 9
      3.1.1 Background and context ...................................................................................... 9
      3.1.2 Technology-enhanced solutions .......................................................................... 11
   3.2 WHAT DID YOU LEARN? ............................................................................................ 12
   3.3 IMPACT ....................................................................................................................... 14

4) CONCLUSIONS & RECOMMENDATIONS ....................................................................... 15

5) IMPLICATIONS FOR THE FUTURE ............................................................................. 16

6) REFERENCES ................................................................................................................. 18

APPENDIX .............................................................................................................................. 19
   APPENDIX A: MODELS REPRESENTING APEL AND ACCREDITATION PROCESSES .......... 19
   APPENDIX B: EXAMPLES OF COURSE DESCRIPTION AND THEIR RELEVANT INFORMATION ARCHITECTURES (REPRESENTED IN UML) ...................................................................................................................... 21
   APPENDIX C: TECHNICAL LESSONS LEARNED ............................................................ 27
   APPENDIX D: THE DESCRIPTION OF THE MUSKET TOOLS ....................................... 29
   APPENDIX E: CALCULATING THE SEMANTIC SIMILARITY .......................................... 33
   Appendix F: MUSKET Tool User Manual
1) Acknowledgements
We would like to thank each and everyone who contributed to the project in various ways. Our appreciation and gratitude especially, to the following organisations, institutions and individuals for funding, participating in, and contributing to the success of the project.

- Joint Information Systems Committee (JISC), especially support offered by the LLL-WFD programme manager Ruth Drysdale, and the members of the SSBR Support and Synthesis team Ellen Lessner, Patsy Clerk and Paul Bailey.
- The project partner QA-IQ for offering access to the curriculum related resources.
- The School of Engineering & Information Sciences for the continuous support given throughout the project’s duration.
- Colleagues from Middlesex University schools who provided feedback and advice.
- Colleagues from the Institute of Work Based Learning who shared their experiences and expertise on lifelong learning.
- Other JISC projects who supported us and gave valuable feedback.
  - Student participants:
    - Taiwo Temitayo Awoniyi of Middlesex University who contributed to the tool as part of his postgraduate project;
    - Noha Saleeb of Middlesex University for her contribution in the evaluation of the project deliverables and the sustainability tasks of the Benefit Realisation phase of the project;
    - TingTing Zeng of Oxford University for her contribution in the collaborative work with the UWIC.
- Richard Staniforth of the University of Wales Institute for his invaluable support, feedback, and knowledge sharing
- The School of Engineering & Information Sciences IT Support team.
2) Report Summary

2.1 Project Overview

MUSKET project aimed to support employer engagement and workforce planning requirements. This was hoped to achieve by developing a software application that can provide an integrated view of curriculum provision from all types of providers such as employers (industrial) and the higher education sector of UK (universities).

The software application delivered two software tools that allow:

- End users to import Word documents containing course descriptions and find the semantic similarity between the documents.
- Employer-led learner route planning – given a number of requirements, to find suitable course provisions.

Looking at program specifications from various universities and our partner QAIQ, it was apparent that HE institutions presented their academic provision in various formats. There are only a few institutions that have a central template for programme and module descriptions. The use of the tool emphasised the importance of having such a template if you need integration with other service providers. During the early investigation stages of the project it was evident that several Universities followed certain templates that were not applicable to all disciplines. It was also difficult to identify the main reasons for the use of such templates and it emerged that certain quality concerns were the main focus rather than the design of a curriculum structure that could be used to describe the main elements of a programme. Similarly the interchangeable use of certain keywords in curriculum descriptions advocated the need for an information model such as XCRI.

Similarity between documents was based on key words selected by the user (semantic mark-up), and the relationship between the keywords. For this purpose it was also necessary to derive the information architecture of the document. The architecture explicitly explains the way curriculum components are linked in a document.

Semantic similarity also takes into consideration the importance the end user puts on each keyword. This enabled us to take into consideration the individual user preferences. The semantic similarity between documents should not be focused on a simple text comparison but an intelligent mapping of curriculum components. Therefore, the semantic similarity is also filtered according different criteria that may affect the user preferences (e.g. which keywords are more important, the context of a keyword, the use of certain wording in the description of curriculum components).

The project as a whole was helpful in looking at the challenges of programme provision from the three perspectives of employer, learner and institution.

Apart from making their programmes available to a wider community, the tool also enables institutions to increase the quality of their programmes offered by highlighting any duplicates and redundancies. The use of the MUSKET tools for an institution could focus on identifying similarities between programmes, ensure that certain differentiation levels are achieved when introducing new programmes and support the comparison with programmes from other universities. For example, if the similarity of two modules in a particular programme, or the similarity of two programmes in a department is very high, this flags up overlaps and duplicates that there may exist.

The tool will enable employers and educational institutions to work together more effectively in a value adding manner. Employers are seeking quality of provision, relevance to business needs and
a delivery method suited to the company rather than the Higher Education Institutions (HEIs). The need for improved communication between HEIs and employers is a desired feature highlighted by the government and many advisory agencies. The semantic similarity between programmes offered at universities and learner providers can assist this alignment. The transformation of in-house courses (e.g. continuous professional development short courses, training programmes) according to XCRI-CAP allows providers to follow a consistent way in presenting their programmes. A different approach in using the semantic similarity is from a learner perspective that offers the means of searching curriculum documents for certain keywords. The learner could potentially assess a number of programmes against a number of keywords and search for the most suitable one according to certain criteria (e.g. a specific topic, key learning outcomes, reading list, placement opportunities).

2.2 Project Outputs

A summary of MUSKET deliverables and outputs (available on MUSKET website www.musket.mdx.ac.uk) is given below:

1. A set of models that represents the accreditation process used in APEL (Accreditation of Prior Experiential Learning) both at the individual and organisational level (Appendix A).

2. Models representing the information architecture of the course descriptions: given in UML, presenting the keywords and their relationships to each other (See Appendix B, for a sample of course descriptions and their relative information architecture diagrams).

3. Deliver two software tools:
   - One that will allow non-technical specialists to import MS Word documents containing course descriptions, provide semantic mark-up, and find the similarity between them;
   - One that will support employer-led learner route planning - Based on the semantic similarity an appropriate learner route can be displayed.

4. Publications:
5. Dissemination activities

- October 8, 2009 – Presentation, WELL Assembly, University of Bradford, Bradford.
- July 2, 2010 – Presentation, MUSKET Assembly, Middlesex University, Hendon, London.
- October 20, 2010 – Poster presentation, SSBR Festival of Assemblies & Trade Fair, Oxford.
- November 17, 2010 – Research Seminar, School of Engineering & Information Sciences, Middlesex University, Hendon, London.
- MUSKET Benefit Realisation Roadshow, 2011
  - January 19 – University of Nottingham (roadshow session)
  - January 20 – University of Wolverhampton (roadshow session)
  - February 2 – Glasgow Caledonian University (roadshow session)
  - February 10 – University of Bolton (roadshow session)
  - February 26 – University of Wales Institute, Cardiff (roadshow session)
  - March 23 – eLearning Experts Meeting, Birmingham (poster presentation)
  - April 7 – Middlesex University (workshop)
  - April 13 – UKAIS conference, Oxford (poster presentation)
  - April 14 – Anglia Ruskin University (roadshow session)
  - June 13-17 – Middlesex University (workshop)
  - June 27 – XCRI assembly, Nottingham (workshop)
  - September 6 – ALT-C conference (poster presentation)
  - TBC May-June 2011 roadmap events

6. Reports

- Interim reports
- Final report
- Dissemination report (March 2011)

7. Co-work in collaboration with UWIC JISC project.

   For example, the team helped UWIC in creating an institutional template to describe their programs that could be easily converted to XCRI, which is now in use.

8. Web presence

- MUSKET website: www.musket.mdx.ac.uk
- Tool Demo: http://www.youtube.com/watch?v=eSaWIq8bbQ
9. Other outputs
   - MUSKET posters.
   - A user-manual explaining how to use the tools (Appendix F)
   - Project/tools evaluation via video, this task was run via road show.
   - BR – Output (following the completion of the roadshow in June 2011)
     i. Case descriptions
     ii. Evaluation results
     iii. Roadshow map and findings

2.3 Impact and Benefits to the Community

MUSKET made a significant impact on all key stakeholders, see Table 1 below.

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>MUSKET’s greatest benefits are…</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Learner</strong></td>
<td>The semantic comparison tool facilitate selection of programmes which are more aligned with individual user needs: There is a vast number of programmes offered on every subject and sometimes there is little variation among some. One of the difficult tasks most learners are faced with is the selection of the programme which best suits their needs. Learning more geared towards required skills rather than what is offered: learners can search for programmes which give them the required skills easily, thus getting desired value from their educational programmes.</td>
</tr>
<tr>
<td><strong>Employer</strong></td>
<td>Alignment of skills gained to employer needs: Employers can search for programmes which will give employees the required skills for a particular role. Involvement in the design of programmes: Employers have a strategic approach to employee training. Employers can benefit by engaging with educational institutions in designing new programmes or enhancing existing programmes especially, in the case of where there is a lack of compatibility between available programmes and required skills.</td>
</tr>
<tr>
<td><strong>Educational Institutions</strong></td>
<td>Help student retention by aligning learner requirements with academic provision: one of the biggest challenges higher educational institutions are facing today is student retention. One reason for student dissatisfaction arises from their inability to select the programme best for them. Institutions can make use of the tool to help students select what is best for them and thus increase student retention. To align with employer’s needs – Higher education is a competitive market today. Educational institutions need to be competitive in not only designing their programmes but also in the way they deliver them. Students also are much more demanding and aware of what exactly they want from their educational programmes. One way of developing competitiveness is by delivering skills and capabilities which are much more valuable to industry than those delivered</td>
</tr>
</tbody>
</table>
### Stakeholder | MUSKET’s greatest benefits are...
--- | ---
 | elsewhere. The tool can be used to monitor skills required by employers. Universities can engage employers in a proactive manner when designing programmes. Agile programme development – quick response to market needs fast track validation process: every industry is in a very dynamically changing environment today. So does the higher education sector. Having the ability to pick modules which satisfy a particular set of requirements, the tool will facilitate more agile programme development. Keeping to quality standards: helps to eliminate delivery of a topic in several modules, thus leading to redundancy in programme delivery; making sure the equality between levels is less, e.g. when the same topic is delivered at both UG and PG level. Supporting APEL claims - By attempting to support APEL claims with their transformation to the XCRI-CAP model, the MUSKET tools may support the comparison of an APEL claim with a number of programme descriptions. This would allow the short-listing of APEL claims and the establishment of a system that filters APEL claims. Possible applications may include the direction of APEL claims to the most suitable programme as well as ensuring that certain APEL decision makers are targeted based on the specialisation of the claim and the skills and knowledge of the assessor.

Table 1: Benefits to stakeholder groups

### 2.4 Main Lessons Learnt

The project success was mainly based on the effective integration of programme/course descriptions within institutions and among various educational providers. Thus many problems faced were much similar to data integration and incompatibility issues. Before any attempts at integrating with external providers, institutions need to standardise the use of terminology and programme/course description formats within organisations. Even within the same institution, programme descriptions are based on different templates depending on certain quality criteria. The use of keywords and the meaning of curriculum components vary significantly between institutions and a different context exists among learning providers.

Conversion of word documents to the chosen standard format required conversion to XML. This process was more complex especially when documents which are in free format were involved. Although there are existing tools to achieve the word to XML conversion, most of these tools lead to some loss of information. Further, interviews with APEL experts and programme designers revealed that comparison of two academic units (programme or module level) usually occurred as a series of comparisons between sections of a document. Hence the transformation process needed to preserve the cognitive coherence between sections as well. This led to the development of a bespoke transformation tool.

The above described transformation process should take under consideration the context for each curriculum keyword used. Most of the established off-the-shelf text comparison tools compare one document to another as a whole but comparing section by section was tedious. The MUSKET tool has an additional benefit of ensuring that the most suitable curriculum components are compared (i.e. the semantic part) and that each curriculum component is checked for specific user provided
keywords that accurately represent the criteria to be used for the comparison of the programmes (i.e. the similarity part).

Automating the APEL process, even partially turned out to be extremely difficult. The APEL process varies from one applicant to another based on their individual requirements. Obtaining sufficient resources in order to understand the logical process behind APEL decisions is extremely difficult. Further, such decisions require synergies between discipline experts academics who are aware of APEL requirements. When programmes are aligned with external courses, the accreditation of external providers and their courses may affect how APEL is performed in institutions. Hence even semi-automation of the APEL process is extremely difficult. More so, such an effort may remove the core of the APEL process, the flexibility it aims to give the learner.

The technical lessons learned are described in Appendix C.

3) Main Body of Report

3.1 What did you do? (Methodology)

3.1.1 Background and context

This project was proposed on UK government’s high level skills strategy aimed at promoting employee development and building a skilled work force within the country. The government strategy also called for change in employer behaviour; not only in investing in higher-level training, but also in creating closer collaborations with HE institutions.

In a survey\(^1\) of 600 companies from all sizes and all sectors in England, one of the most frequently cited barriers for companies to build links with universities, was lack of information both on what universities can offer and who to contact. Most employers were not confident there will be sufficient skilled people available to them to meet their needs in the future.

These issues and the holding of events such as, The Times Higher Education Employer Engagement Conference, ‘How far should universities go to meet the needs of employers?’ drove home the importance and the need for the MUSKET project in 2009.

Although the UK government has changed hands and along with it the government’s strategic aims for higher education, in 2011, when the project is coming to a closure, its importance has still not dimmed. For example, the increase in the tuition fee will increase student demand for more career based skills, and universities will increasingly look to add more ‘Off-quota students’. MUSKET tools can aid in achieving both.

WBL in Middlesex

Work-based learning (WBL) programmes bring together work organisations and the institutions to develop and create new learning opportunities in the workplace, which meet the needs of the learner, contribute to the long term development of the organisation, and are formally accredited as university courses.

In the current environment WBL is seen by many, as a tool by which the industries, companies and nation as a whole can improve their competiveness. As Nixon, et al. (2010) state, WBL “plays an

---

important role in meeting the demand for reskilling and upskilling the workforce”

Middlesex University established The Institute for Work based Learning (IWBL) for the development of work based learning at higher education level about 18 years ago. Today, the IWBL is a nationally recognised Centre of Excellence.

The MUSKET project benefited from the experience and skills of IWBL members who provided a solid understanding of how traditional curriculum development differs from lifelong learning and workforce development programmes. The project managed to obtain support in the form of several procedures used for APEL decision making, accreditation of organisation training programmes and the support mechanism provided for the support of the accreditation process.

Implications

Gallacher and Reeve\(^3\) identify four concepts which will aid in our understanding of work based learning: partnership, flexibility, relevance and accreditation. These four concepts very simply outline the MUSKET project objectives. Aim of the project can in one way be described as: to bring a state of co-opetition among employers and HE institutions, in developing a high-skilled workforce.

The Need for XCRI Standard

For work based learning to be fully integrated and profitable to all stakeholders, educational institutions and organisations in general need to have a common, universal, and standard means of presenting and advertising course related information in order to inform strategic decisions and make adequate and efficient use of resources available. For this end, we use a nationally acclaimed standard of representing course information in United Kingdom called XCRI-CAP (the Course Advertising Profile of XCRI, eXchange of Course Related Information)\(^4\).

XCRI-CAP is basically a XML\(^5\) specification. Learning providers can publish their course information in this standard format, so that course descriptions can be assimilated and searched by service providers such as UCAS\(^6\), hotcourses\(^7\), and Aimhigher\(^8\).

In Work Based Learning a learner should be able to take single courses during his/her work life and carry-forward the credits accumulated. XCRI standard can be used to trace, and document the lifelong learner’s academic career.

The UK higher education industry is not only highly competitive but also highly globalised today. The information provided on our web portals should be easily understood by the learners located

---


\(^5\) Extensible Mark-up Language, See [http://www.xml.com/pub/a/98/10/guide0.html][for a description of XML.

\(^6\) http://www.ucas.ac.uk/

\(^7\) http://www.hotcourses.com/

\(^8\) http://www.direct.gov.uk/en/EducationAndLearning/UniversityAndHigherEducation/DG_073697
both nationally and internationally. This has created a need for powerful information management systems to support the representation, exchange, integration and sharing of curriculum data over the internet. By using one national standard to represent curriculum both in format and in the terminology used we can make searching our programmes easier for globally dispersed applicants. The way the MUSKET tools use XCRI-CAP for identifying the semantic similarity between different courses could be further investigated under the prism of the Bologna process. The alignment of course descriptions could further support transfers between international programmes.

**MUSKET Aims**

The project aims to develop and deliver a software application that will support employer engagement and workforce planning requirements by providing an integrated view of employer based, professional and higher education sector.

The project will deliver:

- Models that will represent the APEL process;
- Two software tools:
  - one that will allow non-technical specialists to import MS Word documents containing course descriptions, provide semantic mark-up, and find the similarity between them;
  - one that will support employer-led learner route planning.

**3.1.2 Technology-enhanced solutions**

Course related information needs to be exchanged between universities, its aggregators, prospective students and third party agents. In order to meet this requirement, this project develops an automated method of transforming course information of various organisations to the chosen XCRI_CAP standard. In order to achieve this, three tools were designed. The design overview of the tools is described in Appendix D.

The first step of this transformation is to convert the documents to XML. This is achieved by the Transform Tool which is described in the next section.

**3.1.2.1 The Transform Tool**

The transform tool is used to extract information from an institutional document and convert that information into XCRI format. The user-perception of the tool (the steps which the user goes through in using the tool is described in the User Manual in Appendix F. Here we give a simple and brief description of the technical performance of the tool.

The architecture of the Transformation Tool is shown in Appendix D1, Figure D1.

Input to the Transformation Tool is a word (or rtf or pdf) document which describes a module or a programme. The document can be free text or in a tabular format and may be internally structured into sections and paragraphs. The paragraphs contain the text that needs to be extracted. The text obtained is then mapped to the relevant XCRI field and validated against the XCRI Schema. First, the document is converted to text format. This conversion removes all formatting from the data in the document, such as font styles, giving pure text which is required for the conversion to XCRI.

The weakness here is the loss of some information. One, the formatting indicates the structure and hence the relationship between different keywords in the document. Second, if the document contains tables or pictures, the software cannot distinguish one from the other. The software will try to transform graphics to text regardless of the type of content of the document, thereby losing
valuable information. In order to overcome these weaknesses we make use of UML to represent the information architecture of the document.

For example, all the keywords, such as, “Course code”, “Overview”, “Fee” will be identified and a class diagram is created. The class diagram not only highlights the keywords but also the relationship among the key words. For example, Appendix B shows three examples of module descriptors and their respective UML diagrams. The rationale behind the use of UML and the details of this part of the transfer is given in Appendix D2.

Information extracted from the text files, and the UML files are combined to arrive at XML files (paths labeled 1 and 2 in Figure D1, Appendix D1). The XML files are then converted to XCRI-CAP standard which is the final output of the transform tool.

The XCRI files are utilised to present a standardised means of exchanging course related information between all parties that an educational institution renders services to: institutions and organizations; institutions and aggregators; institutions and students; etc.

3.1.2.2 The Use of Ontology Modelling

The semantic similarity function of the tool needs to understand the schema and semantics of the XCRI_CAP files, if it is to return informative responses and find the similarity between documents. Ontology plays a key role in realizing this part of the tool, providing a common; shared understanding of the domain knowledge.

Ontology Web Language (OWL) is used to capture knowledge about the domain, which describes the concepts in the domain and also the relationships that hold between these concepts. Appendix D3 describes how we make use of the Ontology Web Language (OWL) in our tools.

3.1.2.3 Semantic Similarity and its Calculation

Taking advantage of the ISA relationship of concepts in ontology modeling, a new method for computing concept similarity was developed. This method combines the semantic distance and statistical properties of the ontology. Semantic Similarity is a value, generally in the range [0, 1]; if two concepts have the same meaning, then the value should be 1, and if no path connectivity within the two concepts, the value is 0.

According to the basic principles of ontology-based similarity calculations several factors should be considered: Semantic Similarity of Property, Semantic Similarity of Instances, and Semantic Similarity of Class. Details of these calculations are shown in Appendix E.

The tool returns a semantic similarity value for each key word (and its architecture) selected by the user and an accumulated value for the whole document is displayed. The case study given in Appendix D3 demonstrates how the OWL files are used in finding the semantic similarity between documents.

3.2 What did you learn?

1. The success of a project depends on a number of factors, some of which are: the support and the contribution from the mother institution; the degree of collaboration from the partner institutions; and the resource availability and accessibility. What is most important is the
skills, the experience, and the perseverance of the project team. Good team work can over
come any adverse conditions faced by the project and bring the project into completion with
positive results. MUSKET, faced negative effects from all the three contributory factors
mentioned above. As a result we could not benefit from the Customer Relationship
Management (CRM) system used by the Institute of Work Based Learning (IWBL) at
Middlesex University. Therefore, the focus of the MUSKET tool was changed from that of
CRM based to document based.

2. If we are to get the best benefit of integration, not only the course descriptions need to be
integrated, but we also need to adhere to a standard set of terminology. The vocabulary or
the terminology used to describe courses varied from one institution to another. Although
this was not a problem for the project team members when comparing documents, for
those who did not have much knowledge of the UK academic systems, especially for those
who did not come form academia, this disparity caused difficulties. Therefore, it is safe to
assume that this must be causing potential applicants, especially those who are overseas,
difficulties in their course searches as well.

3. It was over optimistic of us to expect that we could develop an intelligent component which
could deal with APEL decisions. Automating the APEL process, even partially turned out to
be extremely difficult due to the following reasons:

• The APEL process is very diverse. From one applicant to another requirements
change and hence there is a lack of consistency in the APEL process followed.

• It was extremely difficult to obtain sufficient resources towards well informed APEL
decisions in all disciplines and fields available in a HEI.

• APEL decisions require synergies between discipline experts and IWBL staff aware
of APEL requirements.

• The accreditation of external providers and their courses may affect how APEL is
performed for programmes aligned with external courses.

4. The technically oriented project risks are easier to control/mitigate when compared to social
risks. Most of the unforeseen risks we faced were caused by institutional bureaucracies and
took a lot of time and effort in resolving. In most cases the team had to come up with
alternative solutions.

• The university’s student information system (Middlesex Integrated Student
Information System – MISIS) contains all the programme specifications stored in an
ORACLE database. Hence our expectation was that converting them to XCRI and
comparing them would be easy. In other words we assumed that we already had
our initial test bed of data in hand. Contrary to our expectations it was difficult to
have access to the university database due to heavy security policies surrounding it.
Another reason behind the difficulty to get access was the lack of ownership for the
database. When direct access turned out to be difficult, the team tried to get a copy
of a portion of the database as a test bed. It became soon apparent that the person
who was responsible for creating the database was no longer in employment and no
one was willing to put an effort into providing us with such a copy. This forced us to
create our own database of course descriptions. The first step was to come up with
a tool which can convert Word documents to XCRI, which turned out to be one of
the most useful outputs of the project.

• One of the initial aims of the project was to be CRM based. The project aimed to
benefit from the CRM system the university acquired prior to the submission of the
project proposal. It was discovered that the CRM system aimed primarily to
increase the effectiveness of communication with potential and existing
accreditation clients. Although there should have been more emphasis on
integrating the CRM system with transformation and mapping facilities to allow
harmonisation of courses, collaboration with the CRM team was difficult due to political and proprietary issues. This led to MUSKET deviating from its CRM focus.

- The contact person at QAIQ, left soon after the start of the project. It turned out to be difficult to establish contact with the newly appointed person due to a clash of interest. This highlighted the risk of having high dependency on one single person, even at a senior level of management. This made us dependent on Web-based information alone. The positive aspect is that they provided a different type of input documents for the tool.

If the project success highly dependent on access to certain institutional resources, it is good practice to ensure that access is possible at the time of writing the proposal. Funding agencies such as JISC also can ensure this by making sure that institutional executives do promise to grant their support to the project.

Another way of alleviating this problem is to make decision makers in those crucial areas project partners and thus take some ownership of the project. This may also lead them to make an effort to have a clear understanding of the project aims and goals which may remove some of the political barriers.

5. Getting access to the Middlesex oracle database was a huge challenge as explained in point 3 above. As a result time was taken up creating a database of test data. Therefore, we realised that getting access to test data should have been allocated more resources.

6. Institutions varied in the way they described courses. More unstructured the course descriptions are, the less accurate the output of the MUSKET tool. One way this also highlights a drawback of the tool in its current status which can be enhanced in the future. The tool performs better if the word files are structured.

7. We have realised that the testing and evaluation of the tool requires more effort, time, and man power than anticipated initially. In order to overcome this, a benefit realisation project was commenced with further JISC funding which allowed for further evaluation of the tools. With the feedback gathered the tool continues to develop.

8. The technical lessons learned during development of the tools are discussed in Appendix C.

9. The mapping of an APEL claim to a curriculum development document (e.g. programme description) does not seem feasible to achieve. As a result following the Benefit Realisation phase of the project, the MUSKET tools will be used to prepare an APEL claim towards a PG Certificate in curriculum development. The aim of this activity is to prove that the MUSKET tool can support the preparation and assessment of an APEL claim based on a structured template following certain constraints as set by the use of XCRI-CAP.

### 3.3 Impact

In Section 2.3 we described the perceived benefits of the project to the society.

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>The perceived benefit</th>
<th>The Supporting application/tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learner</td>
<td>Assist learners in the selection of the</td>
<td>The semantic comparison tool facilitates</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>The perceived benefit</td>
<td>The Supporting application/tool</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Institution</td>
<td>Make institutions adhere to a universal way of describing their programme provision. Institutions who want to integrate their data/information with other institutions need to present their course information more structurally following an institutional standard. This helps the institution in their quality control and validation processes and thus can be taken as an indirect benefit. For example, the team helped UWIC in creating an institutional template to describe their programs.</td>
<td>Collaboration with UWIC&lt;br&gt;&lt;br&gt;At UWIC a push for standardisation using the Programme Specification Template and the Module Descriptor Templates in word for new and revalidated courses offered an opportunity for MUSKET to work towards the development of an XCRI compliant template.&lt;br&gt;&lt;br&gt;This collaboration led to the following developments:&lt;br&gt;&lt;br&gt;o Creating a new course template for enabling easier data entry and access for transformation.&lt;br&gt;&lt;br&gt;o Developing a java application for UWIC, using the MUSKET innovative approach through UML mapping and doc2xml to create an XCRI file from existing documents via the new template.</td>
</tr>
</tbody>
</table>

Table 2: Impact of the project

4) CONCLUSIONS & RECOMMENDATIONS

Developing a method to convert Word documents to XML, proved valuable and achieved the goal set out in the project objectives. The MUSKET tool to generate course information in XCRI standard can help HE institution to collaborate more closely and integrate their course provision providing a single platform for end users to search suitable courses. More importantly the tools developed might be shared in the next XCRI related assembly and used by other institutions in United Kingdom to convert their course information in word template to XML.

The algorithm used to calculate the semantic similarity information proved to be efficient. However, further research is needed to make this algorithm more flexible and more accurate. Evaluation has been carried out to compare the results between the current system and other traditional search
The MUSKET tools provided the means to engage in discussions with several Universities that participated in the evaluation of the project outputs. There are several application areas that may require the use of these tools:

- Comparing courses from different Universities
- Comparing courses from the same University
- Aligning commercial provider courses to HEI ones
- Combining HEI and Commercial provider courses
- Accreditation of in house training
- APCL / APEL decision support
- Identifying study routes
- Identifying progression routers
- Providing course locator (most suitable course)

The project benefited by the already existing XCRI standards and the associated resources. The knowledge and experience of the XCRI user community was invaluable during the project. The knowledge sharing culture which is supported and promoted by JISC creates a supporting community among the projects.

During the early stages of the MUSKET-BR roadshow that attempted to demonstrate the MUSKET tools to a number of HEIs, evaluate their usefulness and identify any further applications, a common theme emerged. WBL practitioners and those who were involved with APEL decisions requested a harmonisation of APEL claims and a consistent representation of APEL portfolios. Using XCRI allowed the MUSKET project to align the description of curriculum from different Universities and learning providers. It also provided the means to enable consistency between courses from different disciplines. Transferring credit between different learning organisations is feasible with the use of XCRI-CAP. However WBL programmes are based on APEL decisions that do not follow a standardized structure and description of the learner's experience. Therefore it is not feasible to use the current fields that are defined in XCRI-CAP. We believe that either XCRI need to be expanded to cover APEL claims or APEL claims need to be redesigned to correspond to the XCRI-CAP information model.

The MYSKET tool although adequately functions, has some drawbacks as discussed in the report. The tool can be further enhanced as suggested in Section 5. However, the tool proved that the concept of an integrated portal for course provision as proposed by the project is practical and has numerous advantages. Definitely a single search point for learners who desire to follow any course in UK is a strategically attractive vision.

5) IMPLICATIONS FOR THE FUTURE

Utilize Meta Model Technology into Tool Design

Transformation method presently used by MUSKET involves transforming Microsoft word document to Hyper Text Mark-up Language (HTML). The HTML file is then converted to XML and the XML is mapped to XCRI as shown in Figure 1.
The stages in this process includes

**Stage 1:** First the Musket tool reads the Microsoft word file and converts it into plain text format.

**Stage 2:** The tool checks the text file for keywords based on the UML meta model. For example, all the keywords, such as “Course code”, “Overview”, “Fee” will be identified by the program, and will be imported to a XML file. This design makes the Musket tool more robust, since data from different formats will generate different keywords; after the validation, the text file is transformed into a XML/UTF 8 standard file.

**Stage 3:** The target XCRI file, and the standard format, system will automatically generate the XCRI file base on the semantic meaning from stage 2.

A possible area of research is to make this process more simple and automated. The importance is to capture the essence of the content without losing any information in the transformation process.

**Extracting keywords from course documentation**

Although there are many ways of determining the key words, to easily extract the relationship among the keywords was difficult. This also highlights another area the tool can enhance. If we have a large collection of UML diagrams, it may be possible to arrive at a set of rules using which we may be able to automatically create UML diagrams or at least semi automate the creation of class diagrams. In other words it may be possible to arrive at a basic structure which can be further developed/modified by the user.
More intelligent semantic information systems need to be developed which can understand the schema and semantics of the XCRI_CAP XML files, in order to be able to return more informative and intelligent responses to support learner route planning.

**Allocating weight values:**
Currently, the user needs to allocate a weight to each keyword depending on the priority or importance s/he on each key word. Further research need to be carried out to make this process more robust and standard. For example, the user only indicates the preference hierarchy, based on which the system allocated the weight in a more systematic way.

Currently the tool does not take into consideration accessibility issues. If the tool is to

6) **References**


Appendix

Appendix A: Models representing APEL and accreditation processes

**Figure A1: MUSKET System Architecture**

**Figure A2: Accreditation of professional courses**

**Development of the Proposal**

Following an initial informal discussion the Accreditation Proposal is developed with comment from a University Adviser.

**Stage 2**

**Issue and acceptance of the Contract**

The developed Proposal is submitted to Middlesex University Accreditation Services. The Accreditation Contract is drawn up by to Middlesex University Accreditation Services.

**Stage 3**

**Assessment of the Proposal and consideration by the University Accreditation Board**

Following the second informal review of the proposal the Proposal is assessed. The Assessor’s recommendation is considered by the University Accreditation Board. The Client is notified of the decision of the board.

**Stage 4**

**Implementation of the accreditation**

Middlesex University Accreditation Services produce the Memorandum of Co-operation which governs the use of the Accredited Activity.

**Stage 5**

**Operation and Quality Assurance Monitoring of the Accredited Activity**

The Accredited Activity is monitored informally half way through the activity and formally on an annual basis. Certificates of Credit are issued by Middlesex University Accreditation Services.
- Registering a Recognition & Accreditation of Learning (RAL) module.
- Preparing a portfolio of own learning claim for the area of work with relevant experience in the workplace.
- Estimating level and amount of credits against submitted evidence, supporting the portfolio.
- Indicating the award of certain credits.
- Presenting the individual’s perception of the accreditation experience through a reflective essay.

Figure A4: APEL Process
Appendix B: Examples of course description and their relevant information architectures (represented in UML)

Example 1: QA_IQ Company

a) The Course description

Course Details
Fee £1695
Days 5
Course code APMP-5

Course Outline
PLEASE NOTE: Fee does not include residential costs. Please contact us for the latest costs. Residential courses are normally held in Bournemouth.

Overview
The attainment of accreditations offered by professional bodies such as the Association for Project Management (APM) is increasingly seen as a recognised indicator of competence in project management.

This course provides delegates with comprehensive training to prepare them for the APMP examination (based on the APM Body of Knowledge, 5th Edition). It will enable delegates to understand all elements of the APMP syllabus, get to grips with the specific examination language, and learn strategies and tactics for approaching the examination. Extensive examination practice and feedback is included in the workshop.

Delegates may choose to sit the APMP examination on the last afternoon of the course or at a later date.

This is an intensive five-day course which combines presentations with individual and group practice of examination questions. Delegates will have the opportunity to raise and address examination topics of personal concern and will be expected to undertake some evening work at the end of each day.

Those wishing to take the APMP examination must also book on course code APMPEX-5 (or APMPEXR-5 if re-sitting the examination). Note that examination application forms must be received by the APM at least one week prior to the examination date. You must bring signed photo ID with you on exam days (passport, driving licence, student card) as you will be asked to produce it by the invigilator prior to the exam.

Prerequisites
- This course is suitable for project managers and aspiring project managers. Ideally will have some experience of working in a project environment and have a basic understanding of project management issues
• Delegates will be expected to undertake some pre-course work in order to fully benefit from the workshop.

Delegates will learn how to

• Understand all of the concepts, techniques and processes in the APMP syllabus
• Adopt the best approach to the APMP examination
• Answer essay-type questions on the APMP syllabus

Course outline

Project Management in Context
Project management; Programme management; Portfolio management; Project context; Project sponsorship; Project office

Planning and Project Strategy
Planning and Project success & benefits management; Stakeholder management; Project management plan; Project risk management; Project quality management; Health, safety and environmental management

Executing the Strategy
Scope management; Scheduling; Resource management; Budgeting and cost management; Change control; Earned value management; Information management & reporting; Issue management

Techniques
Requirements management; Estimating; Configuration management

Business and Commercial
Business Case; Procurement; Project life cycles; Handover & closeout

Organisation and Governance
Project Reviews; Organisation Structure; Organisational roles; Methods & procedures; Governance of project management

People and Profession
Communication; Teamwork; Leadership; Conflict management; Negotiation

Examination Guidelines
Examination structure; list, state, describe explain meanings; Question structure; Group and individual examination practice

b) The Information Architecture
Example 2: University of Wales Institute, Cardiff (UWIC)

<table>
<thead>
<tr>
<th>Module Title</th>
<th>Module Number</th>
<th>JACS Subject Code(s) and % of each subject</th>
<th>ASC Category(ies)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems Analysis and Design Techniques</td>
<td>BCO104</td>
<td>G530</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level (0 - M)</th>
<th>Credits</th>
<th>ECTS</th>
<th>Module Value</th>
<th>% Taught in Welsh</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>Taught</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Teaching Period</th>
<th>Pre-requisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term 1</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module Leader</th>
<th>School(s)</th>
<th>Campus</th>
</tr>
</thead>
<tbody>
<tr>
<td>N. Jones</td>
<td>Cardiff School of Management</td>
<td>Llandaf</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assessment Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assessment Type</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>Coursework</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aim(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information system development is an inherently highly complex and difficult task, and criticisms levelled at developers range from over-running budgets, to developing systems which do not meet the business needs they were designed to address. This module is intended to introduce the student to a systems approach to problem solving and use of structured methods for systems development.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>After completing this module the student should be able to:</td>
</tr>
<tr>
<td>Describe data analysis techniques;</td>
</tr>
<tr>
<td>Describe process analysis techniques;</td>
</tr>
<tr>
<td>Distinguish between the various techniques and identify the purposes for which each is used;</td>
</tr>
<tr>
<td>Apply appropriate techniques in the analysis and design of small scale systems.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning and Teaching Delivery Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning</td>
</tr>
<tr>
<td>Tutorials</td>
</tr>
<tr>
<td>Practical</td>
</tr>
</tbody>
</table>
Indicative Content

Need for systems analysis and design.
The life cycle of a system: the stages, activities in each stage, outputs from each stage.
Data modelling techniques: definition of entities and their relationships, approaches to data modelling, including logical and relational techniques.
Process modelling techniques: use of data flow diagrams to include decomposition, the differences between logical and physical DFD’S, and Entity Life Histories

Recommended Reading & Required Reading

Required Reading/Learning Materials

Recommended Reading/Learning Materials

Access to Specialist Requirements
VISIO, Blackboard

---

**Figure A2:** UML model for UWIC Under Graduate course information

**Example 3:** Middlesex University (MDX)

<table>
<thead>
<tr>
<th>Module Code</th>
<th>BIS1200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Title</td>
<td>Database Management Systems</td>
</tr>
<tr>
<td>Credit</td>
<td>30</td>
</tr>
<tr>
<td>Owning</td>
<td>Business Information</td>
</tr>
</tbody>
</table>
Subject  Systems

Aims
This module emphasises the fact that the success of information systems in any organisation is largely dependent on the quality of the underlying database. It is designed to introduce students to the principles of data management and to allow them to gain experience in designing and manipulating databases.

Learning Outcomes
On completion of this module, the successful student will be able to:

Knowledge
1. Describe the fundamental concepts and structures of the relational model and other database models
2. Provide a thorough and systematic treatment of conceptual and logical data design
3. Demonstrate knowledge of SQL by designing, implementing and executing complex queries

Skills
4. Apply relational algebra to extract information from a relational database
5. Apply the techniques of entity-relationship diagramming, normalisation and object-oriented modelling to build database models
6. Use SQL to define and manipulate a relational database

Syllabus
- Database concepts
- Relational Data Models
- Data Integrity
- Relational Algebra
- Entity Relationship Modelling
- Normalisation
- Mapping between ER models and Normalisation
- SQL: (DDL, DML)
- Introduction to post relational data models (Object Oriented Modelling, Object Relational Modelling)

Learning, Teaching and Assessment Strategy

Learning and Teaching
- A one-hour lecture will be used to deliver the theoretical aspects of the module
- A two-hour workshop (seminar/lab) will build on the theory delivered in the lecture and allow student to gain practical experience

Assessment
Regular tests and quizzes, completed in the workshops, will provide the student with formative assessment. The assessed coursework will comprise of two main deliverables:
- Workshop test (learning outcomes 1-6)
- Coursework, including a short presentation (learning outcomes 1-6)

Students need to Pass Overall.

OASISplus will be used to offer problem specifications, lecture slides and general resources related to the module. Students will be encouraged to make use of the online discussion facilities.

Typically, the outcomes of, and feedback on, assessment activities on this module will enable students to record in the PDP portfolio their progress in developing effective learning and communications skills.

Assessment Scheme
100% coursework

Learning materials

Essential

Recommended
Figure A3: UML model for Middlesex Under Graduate course information
Appendix C: Technical Lessons Learned

C1: Transform Process

The software application used to transform the UML to XMI format is ArgoUML. Different applications generate different types of code arguments, vocabulary and instructions. Different applications were tried in creating UML class diagrams, and then creating XML by applying the XSLT code we generated. This process gave us different results to that obtained with ArgoUML. This proved that each application uses different algorithms and different formats to represent the information in the UML class diagram. The XSLT code we generated only applies to the specific elements and code generated from ArgoUML.

This places a restriction on the tool when using the tool for future projects. The positive aspect is that currently ArgoUML is a free, lightweight software which is very easy to use. Therefore, we believe the possibility that it will be used by end users and other projects is very high.

One of the benefits discovered was that, should there be an error in the UML class diagram, which may be detected by the user or any other stakeholder, all that needs to be done is to re-draw the UML diagram. Since the rest of the process is automated the required information will be automatically derived. This is similar to using a spreadsheet. If anywhere in the sheet, data is entered incorrectly, all that has to be done is enter the correct values and the result will be automatically re-calculated.

The aim of MUSKET is to develop a set of tools that is easy to use and can be used by non technical users. When developing the transform tool, we experimented different ways of transforming the Word documents to XML format.

1. One such experiment involved using existing software (SAXON: The XSLT and XQuery Processor9) to convert the UML to XMI format, and then the XMI to XML. The drawbacks of this method were:
   1) this involved the use of two software applications which means for the non-technical user learning two different software,
   2) used UML diagrams to encapsulate all the information contained in the document. Hence this method has a very high degree of participation from the user where the user needs to map all required information from the source document to the XML format
   3) as stated in point 2) the level of automation in the process is very low. For example, in the MUSKET tool only the drawing of the UML and the selection of the key words is carried out by the user. The rest of the process is done by the software.

There are a number of different software to create UML, and there are also different ways to convert the information expressed in the UML diagrams to XML format.

However, as a by-product of the above experiment we achieved another breakthrough that could be useful for XML developers. An extensible style sheet was developed to extract information from the XML files. This was achieved using a trial and error method until the desired information is extracted. An application was written using Java code to extract information from the document using the XML style sheets. Since Java was used the application can be universal and it can be applied to any XML file. The java code can be run independently on any machine and therefore the process itself is independent of the tool.

9 http://saxon.sourceforge.net/#F9.3HE
2. In a second experiment we used XML schema built into Microsoft Word to convert the Word files to XML. The XML schema is freely available on the XCRI website (XCRI-CAP v1.1) to be downloaded.

First, the file to be transformed is opened using Word. Then, the schema to be used need to be selected. The next step is to associate the relevant content to the appropriate XCRI element. Since this step can be done easily (in a visual manner) the need for keywords is eliminated here. After selecting the content to be transferred the word file is saved as an xml file. The saved file will now be in XCRI format.

Although the process sounds simple, this again turned out to be a too technical process than that we desired.

The drawbacks of this method are:
1) This method can only deal with files that can be opened with Word (2007 and onwards, in other words need the ‘developer’ function to be able to be installed), for example, does not cater for pdf documents.
2) The user needs to be a technical person who understands what an XML schema is. The user also needs to select the XML options to be used with the schema. For this the user needs to have a clear idea of the available options and what to be used.
3) The user needs to be familiar with Microsoft Word XML Transformations, i.e. loading the schema and mapping the contents of the word document to the schema.
4) The loading of a schema in word and mapping of the contents to the XML Schema has to be repeated for each word document. This is because each document has different delimiter names, different order of delimiters or no delimiters for different contents present in the world file. This is true even for files having the same template.

C2: UML tool

In modelling, the created model is based on the perception of the modeller. The models created may vary depending on who creates them. Consequently the results derived from them may vary as well. The goal is to have a complete representation of the information presented in the document. Omission of any may result in the tool giving an incorrect answer.

Another major issue in modelling is trying to find the right associations. For example, a course is related to many other objects, like the campus, the school, the course leader etc. These relationships are sometimes not one to one, and depend on the nature of the institution. For example, some institutions have several campuses, each offering the same course. Without significant internal knowledge of the system being modelled, it is difficult to find the relationships among the information objects. Modelling is an area of the project that is completely enacted by the user. We do not believe a software tool can cope with modelling the characteristics, behaviour and inbuilt assumptions of the system being modelled without extensive intelligent capabilities.
Appendix D: The Description of the MUSKET Tools

Appendix D1: The Transformation Tool

The Transform Tool converts a word (rtf or pdf) document to XCRI by several steps as shown in Figure D1. Labels 1 and 2 denote the two different paths via information is gathered for the creation of XML files. Label 3 denote the application of rules in matching key words to XCRI fields. Label 4 represents the validation of the XCRI output through the XCRI schema (obtained from the XCRI website). The validated XCRI files are stored in a repository (label 5 in Figure D1).
Appendix D2: The use of UML

The Unified Modeling Language (UML) is a well accepted modelling language which helps to specify, visualize, and represent systems, including their structure and design, in a pictorial form. UML was chosen based on its wide usage. The class diagrams of UML can be used to show clearly and concisely the information architecture of a document.

ArgoUML was used for this purpose. This particular version was selected due to its ease of use and free access. ArgoUML, also has the ability to export the UML class diagrams into an XML Metadata Interchange (XMI) XMI format, which facilitate the conversion to XML. The drawback is the XMI files automatically created from the UML diagrams contain far more information than that required for our purpose. Hence a process of garbage removal needed to be devised to extract the required information from XMI files.

For this purpose a Java function was developed which extracts the useful classes and their attribute information from the XMI files.

Appendix D3: The use of OWL

Ontology is a richer language for providing more complex constraints on the types of resources and their properties. Different ontology languages provide different facilities. The most recent development in standard ontology languages is Ontology Web Language (OWL) from the World Wide Web Consortium. In MUSKET, OWL is used to capture knowledge about the domain. OWL describes the concepts in the domain and also the relationships that hold between the concepts.

We used Protégé by Stanford University (http://protege.stanford.edu/overview/) for constructing the ontology. Below is the description of how Protégé 4 is used to create the OWL files.

- **OWL Classes** are interpreted as sets that contain individuals (see Figure D2). Individuals are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class.

- **Properties**: are similar to relations in UML (and other object oriented notations). For example, the property “belong to” might link the individual “Learner 1” to the property “Role”.

- **Individuals** represent objects in the domain. OWL does not use the Unique Name Assumption (UNA) that means two different names could actually refer to the same individual.

![Figure D2: Classes, Relations and Individuals within Protégé 4](image-url)
After Modelling the course data in OWL, we apply the algorithms described in Appendix E to the models using a knowledge representation tool which accommodates semantic relationships and similarity of course data. In order to demonstrate this, a case study in information retrieval is presented below.

Say, a user wants to compare a course description from Bolton University to one from the QA-IQ company.
- Assume that OWL files have already been created for each of the courses from Bolton University.
- The Transform tool imports the rtf file representing one of the courses provided by the QA-IQ company into XCRI-CAP.
- The semantic tool will convert the XCRI file to an OWL file.
- Using the Semantic tool we import the owl files from Bolton university. Each Bolton University file will be compared to the owl file from QA-IQ company. In other words semantic similarity values of class, instance and property between the two owl files will be compared. This process is shown in Figure D3.

![Figure D3: Finding the semantic similarities of different owl files](image)

Usually the learning outcomes is the key factor used by users to compare the similarity of different courses. Figure D4 shows the results of the mapping for this key attribute given by the system.
This way the system allows non-technical specialists to import unstructured documents containing course descriptions, provide semantic mark-up, and view the semantic similarity between the submitted documents. The similarity index will also support employer-led learner route planning.
**Appendix E: Calculating the Semantic Similarity**

We describe below how the semantic similarity is calculated for a document.

Each class has its own instances and properties. The system will set different weights for instance, object property, and data property. How the semantic similarity of Class is calculated is described below.

**E1: Semantic Similarity of Property**

*Value of Object property:*

\[ S \alpha = S_d + \alpha (1 - S_{ed}(N_A, N_B)) \]

*Value of Dataproperty:*

\[ S \alpha = S_d + \alpha (1 - S_{ed}(N_A, N_B)) \]

\( S_d \) is the Semantic Similarity (SS) value of domain class, \( S_r \) is the SS value of range class; \( \alpha \) and \( \beta \) are the weight value, which are between \( [0-1] \). \( S_{ed}(N_A, N_B) \) is an edit-distance string matching algorithm: Levenshtein, where \( N \) is the name of the property. The string edit distance is the total cost of transforming one string into another using a set of edit rules, each of which has an associated cost. Levenshtein distance is obtained by finding the cheapest way to transform one string into another. Transformations are the one-step operations of (single-phone) insertion, deletion and substitution. In the simplest version substitutions cost about two units except when the source and target are identical, in which case the cost is zero. Insertions and deletions costs half that of substitutions.

**E2: Semantic Similarity of Instances**

Instances are the individual descriptions of each class. For example, the Learning Outcomes of a module can have three instances, where each instance is expressed using a sentence. The Musket Tool needs to compare each sentence with the Learning Outcomes of the other module being compared. The algorithm used here is based on Cosine Similarity\(^\text{10}\) and a user allocated Weight given to the terms in the document selected for comparison. The allocated weight differs for different documents. In other words, the same term may be allocated a different weight in different documents.

Algorithm for calculating semantic similarity of instances is given below.

For instance, if \( D \) denotes the document, and \( t \) denotes the allocated weight, and \( n \) denotes the maximum number of terms in that document, then the weight of the document \( k \) is given by:

\[ D(t_1, t_2, ..., t_k, ..., t_n), \quad 1 \leq k \leq n, \]  

where \( k \) is the number of terms considered.

For example, comparing two documents, Learning Outcomes from Middlesex University and QA company, the system will set different weights for different terms, such as Java, Database, etc. To calculate the value of different \( tk \), the formula is:

\[ \text{http://en.wikipedia.org/wiki/Vector_space_model} \]
E3: Semantic Similarity of Class

Each class has its own instances and properties. The system obtains semantic similarity by setting different weights for $S_1$ (instance), $S_2$ (object property) and $S_3$ (data property), the formula used are shown below, where $\alpha$ and $\beta$ are weight values between [0-1].

$$s = \alpha \cdot s_1 + (1 - \alpha) \cdot (\beta \cdot \sum_{i=1}^{n_1} \frac{n_1 \cdot s_2}{n} + (1 - \beta) \cdot \sum_{i=1}^{n_2} \frac{n_2 \cdot s_3}{n})$$