

BioTune, a component based repository of BioInformatics teaching modules.

Q. Mubarak¹, J. Dmitrieva¹, T. Vermeulen¹, Y. Zhou¹, L. Groenewegen², S. Basmagi³, R. van Beek⁴ and F. J. Verbeek^{*,1}

¹Imaging and Bioinformatics, Imagery & Media Group, Leiden Institute of Advanced Computer Science (LIACS) Leiden University, the Netherlands

²Software Engineering Group, LIACS, Leiden, the Netherlands

³Microbiology, Wageningen University, Wageningen, the Netherlands

⁴Netherlands' BioInformatics Centre, Nijmegen, the Netherlands.

Keywords BioInformatics; E-learning repository; Metadata.

Development of course modules requires a significant amount of time, especially if it is required to be available on line. A curriculum contains "must be thought" material and to develop this in an e-learning like environment feels like reinventing the wheel. This is especially the case for relatively young teaching areas like Bioinformatics; a multi disciplinary field of research including molecular biology, computer science and mathematics.

Teachers can teach each other and learn from experiences in composing learning modules. In the Netherlands, a consortium of Universities and Polytechnics have joined forces in developing a repository of Bioinformatics courses that are made suitable for exchange so that all institutions involved in Bioinformatics education can benefit from fruitful efforts of others. The repository is initially aimed for teachers involved in Bioinformatics, but it can however be adapted to be accessible to students. The organization and means of exchange of educational material can easily be transferred to other areas of research education.

In this paper we explore the development of such a repository, the advantages and its usage. The content in the repository has to be shareable, manageable and reusable. These requirements are translated to an implementation on the basis of a Content Management System (CMS). To be able to represent a broad range of Bioinformatics courses, the structure of the repository must be tailored to a generic framework for polytechnic and university courses. In addition, also the variety of multi-media content types should be well represented. The CMS should adhere to international standards; for this project MediaSurface 5.3 was used.

1. Introduction

In the past decades Bioinformatics has significantly gained importance in the life sciences. In The Netherlands huge priority is given to Bioinformatics curricula at various levels of education. In order to facilitate students from different backgrounds courses need to be developed and accustomed to the backgrounds of the students. To this end the need for a content management system is required where teachers are able to compose learning modules based on the backgrounds of the students and are able to learn, share, and use information provided by other teachers. There is no unified repository where this can be accomplished at the moment. To help develop course material and content in Bioinformatics, we developed a system to help facilitate to share exchange and reuse the content. The idea is to provide a repository which is a unified platform where all the courses and content are shared and which is accessible to everyone. In this paper we will explore the development of such repository called BioTune. We will

* Corresponding author: fverbeek@liacs.nl www.liacs.nl, Phone: +31 71 5275773

explain the different processes that content goes through before being uploaded and the main parts of the BioTune repository.

The current research and development in the field of Bioinformatics repositories is either restricted to database management [1] or image repositories [2]. While Bioinformatics repositories for teaching materials have been developed in the past, their either restricted to local teaching materials or are not designed for development of Bioinformatics course modules [3]. Teaching repositories in other fields such as mathematics or computer science do exist. Users can browse through these repositories and search relevant material through them [4]. Similarly other interesting research in other fields is a virtual knowledge marketplace for multimedia teaching objects [5]. It allows teachers to trade knowledge material between each other. Other web based learning environment is a project called Connexions. It contains material for students, professionals and teachers. Everyone is able to use and reuse the content which is organised in small modules which can be connected into larger courses [6]. Comparable efforts have been made in other multidisciplinary fields like Artificial Intelligence. For Bioinformatics, a repository in which teachers can learn from each other and compose “new” learning modules has not been presented to date. This paper is structured as follows. First the design of BioTune is discussed with respect to both the repository requirements as well as the coordination requirements. Next we continue to describe the implementation on basis of the design and here we introduce XML and the CMS of implementation. We end with a brief discussion of the results so far.

2. Design of BioTune

The data that make up a course content can be variable; i.e., documents of various formats, pictures, movies and/or sound fragments. A data model should cover this range of content and therefore be available first. In addition course content should be dissected in such a way that individual elements, i.e., learning objects can be addressed while at the same time a hierarchy structure in the content is currently maintained.

2.1 Data model

The rationale of the BioTune project is that of Learning Objects (LO). Learning objects are, according to the definition given by IEEE: “Learning objects are entities, digital or non digital which can be used, reused or referenced during technology supported learning.” In the context of BioTune it is an autonomous entity which can be used and reused by teachers to compose material. Learning objects can be of two types:

- Container Item, which can contain other LO entities
- Simple Item, Container which cannot be divided any further.

The container items are organised in a hierarchy: course, module, chapter, section, paragraph etc. According to this hierarchy, a course is a container containing modules and chapters, and a text item is a Simple Item which contains only text. In the same way graphics items are added.

2.2 Paradigm

For a strategy of content re-use we studied behavioural coordination of the system using Paradigm, a coordination language [7]. Using Paradigm modelling language led to a better understanding of the system’s behaviour. We defined different types of users each having their roles in the system. This is illustrated in Figure 1, which shows the different roles. This ensures that users have access according to their use of the system. According to these roles an author can create content, reuse content or delete content, whereas a user can only reuse content. An author whom has created content, another user whom wants to change that content has to submit it, through the reviewer or the committee. An editor can modify new content, but only a person in the committee can delete content. These roles are needed to ensure there are no redundancies, duplication of course material in the system or deletion by unauthorised persons. An

administrator is the person who has the entire authorisation over the system. He can give different rights to the different users and control their access.

The main question about reusability of course material is, what will be done about the course items that another teacher wants to use and modify? There are three possibilities we have in consideration right now.

- Users make a copy of what they need and change it where they want.
- Users change the course items they need and use it in making their own courses.
- If users want to use an item in the CMS in their own courses they could get an instance of the item. Each reusable item will get the number of instances, which would be equal to the number of users whom have used it.

The first option is tedious for the users; since they would have to find the item, copy it to their own content and then use it. The second possibility leaves the issue of users changing each others content. Original contents could be modified without the author's approval. In the CMS the item will be the same. This problem could be solved by giving rights to who can modify the content or not. The third option is the most likely choice for implementation. Users can change or reuse items and the item will be saved as a different item in the CMS.

3. Implementation of BioTune

Prior to exist as autonomous hierarchical components in the CMS, a course has to be inserted in the repository. The submission to the CMS is accomplished in two main steps. In the first stage the course is parsed to produce a generic XML file. This file can be uploaded into the system in the second stage. After upload users can chose to add metadata to the course. Once the course is uploaded users can view the course and use it. To ensure that people using the system contribute to it as well, anyone who uses the system is required to contribute to it. This process is show in Figure 2. The user supplies the courses and can add metadata to the course. Administrator takes care of parsing, uploading and adding metadata as well. This process and the different parts of the BioTune repository will be explained in more detail below.

3.1 Parser

The first step course content passes through is the parsing to a generic format. Course material is available in a range of formats and, therefore, a conversion to a generic structured file is required. This intermediate format is realized through an XML-file that embodies the generic course architecture as a schema. The conversion to the XML file is accomplished via parsers specifically designed and implemented for this goal; i.e., input documents in word, PDF or html form are automatically converted in that manner.

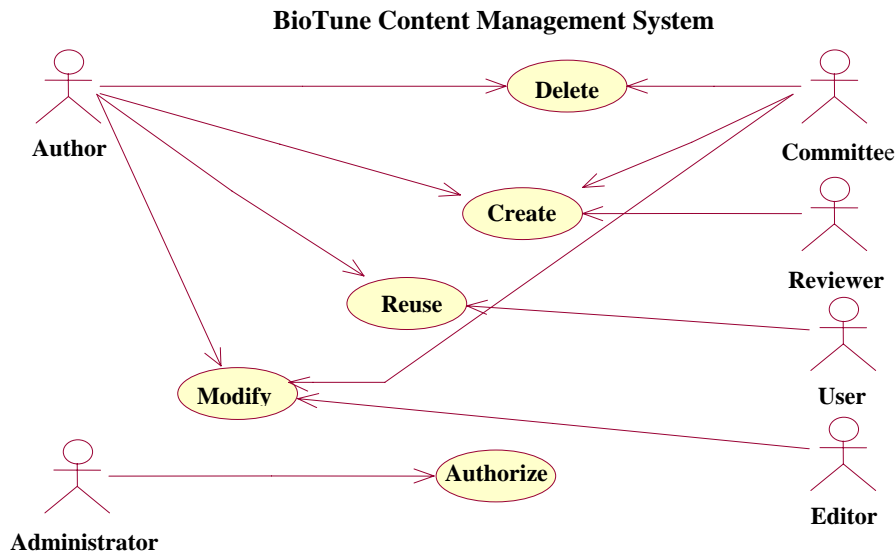


Figure 1, System behaviour modelling after [8]; each ellipse indicates actions allowed (roles) by player. Each of the puppets portrays a role in the system.

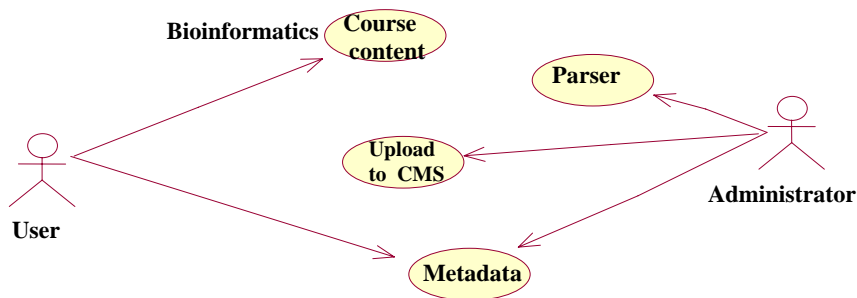


Figure 2, BioTune course upload workflow; the administrator is required as moderator.

3.1.1 Word format

The parser that we have written uses the Jakarta POI Java library [jakarta.apache.org/poi/], which provides access to files in the Microsoft Word 97/2000/XP format employing an API. Most layout elements like tables and simple text mark-up are extracted from the word file and we create CDATA in our XML containing HTML-code with this layout. Since we are mainly interested in the content, the information about fonts and letter sizes is discarded. To determine the structure of the word file, i.e. chapters, sections and paragraphs, the style sheets defined in the file are utilized. We assume that the creator of the word files has been stringent in giving all section titles a certain style. Consequently, at the moment a piece of text that seems to be the title of a chapter is found, all content between that title and the next will be put in one chapter.

3.1.2 HTML format

For a simple HTML file a couple of actions are undertaken. First, tags and attributes that we don't need are removed. These are mostly references to style sheets. Next, all images and links to non-html files are

dealt with. These images and files are retrieved and stored in the CMS, and the XML file is given specific elements to refer to these images and files. Similarly the word-parser titles in the HTML code are handled; i.e., recognised by <h3> tags.

In addition, our software is able to quickly parse structures of many HTML files into one XML file. The parser is given one HTML file, parses its content, but also recursively parses all files that are linked to this file, into the same XML file. Circular links are prevented by keeping track of all the files parsed. At this point for each course a specific parser was written, in this manner acknowledging the structure in the course files. This strategy is preferred as there are many differences in course structures.

3.2 XML Schema

In order to map the course content onto the content management system, we developed an XML-schema for course content. The specific schema representing the generic course layout is strictly adhered to during conversion to the intermediate XML file. The parser ensures that all the different documents that come in different formats and styles once converted adhere to this schema. This is important since the same schema is present in the CMS. Therefore if a course has to be uploaded the XML file of it must be properly converted.

BioTune courses in the XML schema follow a simple grammar defined for this purpose. The courses which are uploaded follow a hierarchical representation in the XML schema. A course consists of module, modules of chapters, chapters of sections and sections of paragraphs. One course may contain one or more modules and similarly one module can consist of at least one chapter. Recursion was introduced into the system so that an item can be defined by itself. This is needed to ensure there is no stiffness in the schema. Users can make constructions such as sections containing (sub) sections etc. The resulting XML file passes to step 2 in which it is uploaded. An important advantage of using the intermediate XML-file is that we are not exclusively depending on one particular CMS. The system could easily be transported on any other platform.

3.3 Media Surface CMS

This basic design is matched in the content management system of BioTune as small information types. The concept of LO is needed for reuse of the different types. These types are stored in the CMS so that they can be reused.

In the second step the separated components from the XML-document are inserted in the CMS repository. This is accomplished by uploading the XML file onto the system. Within the CMS a skeleton framework including the same schema of the XML file of a course layout is embedded. XML items in the CMS are of two types; i.e., branch or leaf. Branch items consist of one or more leaf items or other branch items; e.g. modules. Leaf item does not contain any other items or cannot be divided any further like text items. This structure reflects the hierarchical model of the course and the idea of the LO. It also gives the possibility to make a distinction between the weak related concepts and strong related [9]. At the moment the material that is used or viewed, the different types of it are reassembled on the basis of their metadata, and after submission metadata is added to the course material.

3.4 Metadata

After submission users can opt to add metadata to the course directly. Metadata is the data describing the data or information known about the data in order to provide access to the data. There are four different types of metadata: i.e.,

- Descriptive :title,author etc
- Administrative :how stored,rights etc
- Structural :relation between the information types
- Content :keywords

We are using descriptive and content metadata for now; when the system is used by a broader audience administrative metadata will be added. Metadata is needed for the easy retrieval and re-use of data. The metadata are both general, i.e. Learning Object Model (LOM) related, as well as specific to the domain.

The domain specific metadata are extracted from ontologies and MESH terms. The creation of ontology context for course components enables to see CMS content as part of the whole BioInformatics domain with all existing relations between concepts in ontologies.

The metadata can be added in the form of a XML file. The metadata XML file contains the main keywords related to a course. Even though we adhere to the LOM standards for the metadata, we have not used all the fields described in the LOM metadata model. Our metadata file contains four main descriptors: Author, title, identifier and keywords. This file can be uploaded into the system. The data in the XML file will be added to the metadata item of that specific item in the CMS. This way we ensure there are no redundancies and the metadata matches the metadata of the CMS. Users can also opt to search for the item in the CMS and then add metadata to it. Upon retrieval of the metadata item of a specific item, it will be converted in the form of a XML file again. We continue to use XML to be independent of the CMS. [9]

3.5 Retrieval

Specific queries can be built in order to search through the course material in the CMS. The core of the system is a CMS to which teachers can upload their courses, compose new courses through rearranging existing courses, exchange information as well as retrieve information. Retrieval of components is the pivot of these actions. Retrieval is implemented on basis of metadata annotation of each course. The course metadata contains structural information (e.g. a chapter contains specific paragraphs), learning objects specific LOM metadata, as well as content described through ontology metadata, and therefore a comprehensive query engine can be built. The search could be done according to a set of predefined queries that can be composed on the basis of metadata. These queries can show what other possibilities or combinations are possible with the data in the CMS [9]. Users are facilitated in their search of data in the CMS. Since the system is intended for a wide audience, retrieval and enhancement of the retrieval of data is important.

4. Results and Discussion

The repository is presented in the form of a website; at the moment an α -release is available to the prime users. This version will be invoked in the testing and evaluation phase before further development continues. Large scale testing will be applied; including more stakeholder interactions such as uploading and retrieval of course content. At the moment uploading is accomplished through the administrator who at the stage of the development is equal to the implementer. As the project progresses the implementer and administrator role are separated such that it will be according to the roles described in our coordination model. So the current website is an artefact for demonstration purposes so that teachers can access their course.

We have added course content from different universities and polytechnics to the repository and the system is expanding continuously. Ultimately, we have planned to let teachers parse and upload the courses themselves. Those who are interested should look at [bio-imaging.liacs.nl \(projects\)](http://bio-imaging.liacs.nl/projects) or contact the project management.

As far as the initial parsers are concerned, some problems arise with API libraries because of incomplete code and poor support. Notwithstanding, we are planning to build a generic parser for all courses, which inevitably will be less specific and accurate. The XML2CMS parser is a separate program and has its own different interface. Since the purpose of the parser is conversion of course material we kept it as a separate program from the repository.

In the future the system could be further extended to enable students to study BioInformatics so as to expand their knowledge in specific areas, i.e. where shortcomings exist. The latter is derived from a profile describing their curricular background. The generic aspects and workflow of this system are easily transferred to education programmes in other scientific areas.

Right now the system is used by a group of universities giving their courses for uploading in the system and contributing to it. In the future the system could be expanded to a broader group and be accessible to

teachers interested in contributing to it. As mentioned before the system has not been tested on a large scale. Once there is enough data in the repository, teachers contributing to it will be asked to upload their material themselves. This way we could test the usability and the working of the system.

5. References

- [1] C.J. Robertson, The Forgotten Issues of Bioinformatics Repository Management. (1999)
<http://www.csd.abdn.ac.uk/publications/TR/1999/9901.html>
- [2] A.J Warner, B.D. Athey, M. Pao, W.B. Panko and J. Holden, A Network-Based Image Repository for Biomedical Researchers. 18th Annual Symposium on Computer Applications in Medical Care (SCAMC). (1994)
http://www.generationcp.org/vw/Download/Commissioned_Research_2005/CB4_Dev_of_bioinf_course_mat.pdf
- [3] S. Pongor, Biological Research Centre, University of Hungary. (2006)
<http://www.szbk.u-szeged.hu/modules.php?name=Sections&op=viewarticle&artid=50>
- [4] Repositories for Math and science activities, <http://paws.wcu.edu/emcnelis/MATH693/Repositories.html>
- [5] M. Engelhardt, A. Kárpáti, T. Rack, T.C. Schmidt, A Virtual Knowledge Market-Place, EuroDL . (2001)
<http://www.eurodl.org/materials/contrib/2001/icl01/schmidt/ICL2001-paper.html>
- [6] R. Baraniuk, S. Burrus, B. Hendricks, G. Henry, A. Hero, D. Johnson, D. Jones, R. Nowak, J. Odegard, L. Potter, R. Reedstrom, P. Schniter, I. Selesnick, D. Williams, W. Wilson, Connexions: Education for a Networked World, in IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'02, Orlando. (2002)
<http://cnx.org/>
- [7] L. Groenewegen, E. de Vink, Operational semantics for coordination in Paradigm, In: F. Arbab, C.Talcott, editors, Proc. Coordination 2002, LNCS Volume 2315. (2002)
- [8] Y. Zhou, Coordination Modelling and Design of Adaptive Learning Content to Content Management System: PARADIGM Modelling in the Case Study of BioTune Project, Leiden University Netherlands, LIACS Technical Report. (2006)
- [9] J. Dmitrieva, F. Verbeek, Technical Report project BioTune, Leiden Institute of Advanced Computer Science. (2006) [http:// bio-imaging.liacs.nl](http://bio-imaging.liacs.nl)

6. Acknowledgements

We would like to acknowledge our consortium members for their participation with course material and their comments on the design and implementation of the BioTune platform. Specifically, Dr. C. van Gelder, Prof. G. Vriend (Radboud University Nijmegen), Dr. P. Schaap (Wageningen University of Agriculture), Dr. P.J. Nap (Hanze Hogeschool Groningen) and Prof. A. Siebes (Utrecht University). Further we would like to thank Drs. J.W. Tellegen and Drs. M. Heijkamp for their support in the initial phase of the project.