Proceedings of I-KNOW ’06
Graz, Austria, September 6 - 8, 2006

Visualizing Reuse: More than Meets the Eye

Joris Klerkx
(Katholieke Universiteit Leuven, Belgium
joris.klerkx@cs.kuleuven.be)

Katrien Verbert
(Katholieke Universiteit Leuven, Belgium
katrien.verbert@cs.kuleuven.be)

Erik Duval
(Katholieke Universiteit Leuven, Belgium
erik.duval@cs.kuleuven.be)

Abstract: In this paper we discuss an interactive visualization application that aims to visualize a large repository of small reusable content components that were created by disaggregating legacy content. The purpose of this decomposition is to produce content that can be automatically reused in on-the-fly assemblies of new learning objects. The purpose of the visualization application is to offer insight in the structure of the contents of the repository and to enable access to them in an effective and efficient way.

Keywords: learning objects, reusability, learning object repositories, metadata, information visualization
Categories: H.3.1**,** H.3.3**,** H.3.7

1 Introduction

Learning objects (LOs) are often very coarse-grained and difficult to reuse due to the fact that they are stored in a final presentation form. This static representation is not suitable for flexible content reuse, as the components cannot be easily accessed. In many cases, paragraphs, images or diagrams are assembled manually by copy and paste actions. In order to support this process in a more methodical way, we decompose composite learning objects, and make those components available for more flexible content reuse. This process results in a Learning Object Repository (LOR) with a large amount of LO components and requires advanced support for searching and finding relevant components.

In [Klerkx, et al., 04] we covered our initial research on novel access paradigms for LORs, through Information Visualization techniques. We described the use of tree-maps, hyperbolic trees and Venn diagrams to visualize LOR contents. In [Klerkx, et al., 05], we presented our open and extensible information visualization framework. This framework was developed to support our research by enabling us to rapidly experiment with:

- case-studies that rely on a variety of data sources, with a variety of metadata schemes like IEEE LTSC Learning Object Metadata (LOM), Dublin Core, etc.

- visualization techniques, like tree-maps, radial trees, fisheye views, zoomable user interfaces, etc. [Card, et al., 99].
- information visualization toolkits like Piccolo [Bederson, et al., 04], INRIA's Infovis [Fekete, 04], the InfoVis Cyberinfrastructure [Baumgartner, et al., 04], etc. as we don't want to reinvent the wheel and prefer to combine the strengths of several toolkits.

In this paper we discuss an interactive visualization application that was created by using this framework. The specific application discussed in this paper visualizes a repository of small reusable content components. These components were created by disaggregating legacy content. The purpose of the decomposition is to produce content that can be automatically reused in on-the-fly assemblies of new learning objects – see section 2.

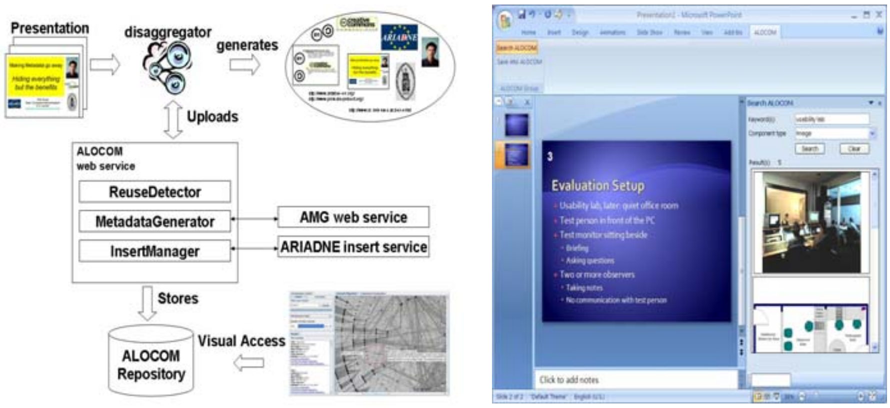
This paper is structured as follows: in the next section, we offer background on the data source and its objectives. The visualization application itself is discussed in section 3. In section 4, we elaborate in a discussion about early evaluation and future work. We conclude the paper in section 5.

2 The ALOCOM framework

The ALOCOM framework supports two processes: the decomposition of learning objects into their components (text fragments, images, definitions, diagrams, tables, examples, audio and video sequences …) as well as the automatic assembly of these components in real-world authoring tools.

- In the disaggregation process, a composite learning object is decomposed into its components. In the case of a slide presentation, the presentation is disaggregated into the individual slides and each slide is further decomposed into its images, diagrams, tables, text fragments, definitions, references… Metadata is added to each learning object component and the component is stored in a repository [Verbert et al., 2005].
- The aggregation process searches for components in the learning object repository and adds them to the learning object. Since users prefer to use authoring environments they are familiar with, this functionality has to be integrated in ordinary authoring tools. We have currently developed a component search plug-in for MS PowerPoint (Figure 1b) and MS Word. A user can specify the type of component he or she is interested in (e.g. slide, image, text fragment, table, diagram…), as well as keywords that best describe the component. All components that satisfy the specified search criteria are shown and the author can easily incorporate them into the learning object he or she is working on. For more details, we refer to [Verbert et al., 2005].

In the remainder of this section, the decomposition process and the ALOCOM repository are described in more detail.



a) b)

Figure 1: The ALOCOM decomposition framework (a); the ALOCOM plug-in for MS PowerPoint (b)

2.1.1 The decomposition process

Decomposing learning objects is a complex task that does not only involve extracting components from existing learning objects. Similarity measures are needed to detect reuse at the component level, and to avoid (near-) duplicates in the repository. Furthermore, metadata need to be automatically added to each individual component, taking into account information from the original learning object to which the component belonged. Relationships between components need to be stored in order to keep track of reuse.

We currently support decomposition for MS PowerPoint presentations (Figure 1a). In a first step, a presentation is parsed in order to retrieve its content and structure. We developed a .Net dis-aggregation client that extracts this information. Furthermore, the presentation is physically decomposed into its components. Each slide is stored in the MS PowerPoint format, images are extracted and text fragments, tables and diagrams are represented in a corresponding XML format.

In the next step, each individual component is uploaded to a dis-aggregation web service, which provides the following functionalities:

1. Reuse is detected using simple metrics that compute similarities between learning object components. The cosine similarity measure is used to detect overlaps in text fragments, hash functions are used to detect identical images, and a combination of these techniques is used for detecting similar slides, presentations, tables and diagrams.
2. Metadata is generated for each component using the Automatic Metadata Generation framework (AMG) [Cardinaels, et al., 2005]. Additional content extracted by ALOCOM disaggregators is added to title and description fields.
3. Learning object components are stored in the ALOCOM repository using the ARIADNE insert service [Ternier & Duval, 2003]; an identifier is generated,

"isPartOf" and "hasPart" relationships between components are stored and the identifier is returned to the disaggregation client.

2.2 The ALOCOM repository

The ALOCOM repository is currently filled with 48286 components that were extracted from 653 presentations. These components include 14113 slides, 5768 images, 198 tables, 26 diagrams and 27543 text fragments. The average reuse in this data set is 0.22. The reuse is calculated as follows for each presentation. the number of reused components in the presentation is divided by the total number of components in that presentation. For instance, a presentation that contains 10 components, among which 2 images that are reused in other presentations, will have a 0.2 reuse value.

3 Visualization of the ALOCOM Repository

In this section of the paper, we will elaborate on the use of information visualization techniques to access the ALOCOM Repository. This visualization application was created to offer the users:

- a general overview of the content components in the repository, their reuse and how they are put together in aggregate content,
- the possibility to easily see if and where the components that they created themselves are reused in aggregates,
- access to reusable components by having the ability to interactively filter out components of their interest and enabling them to reuse these components in their own aggregates.

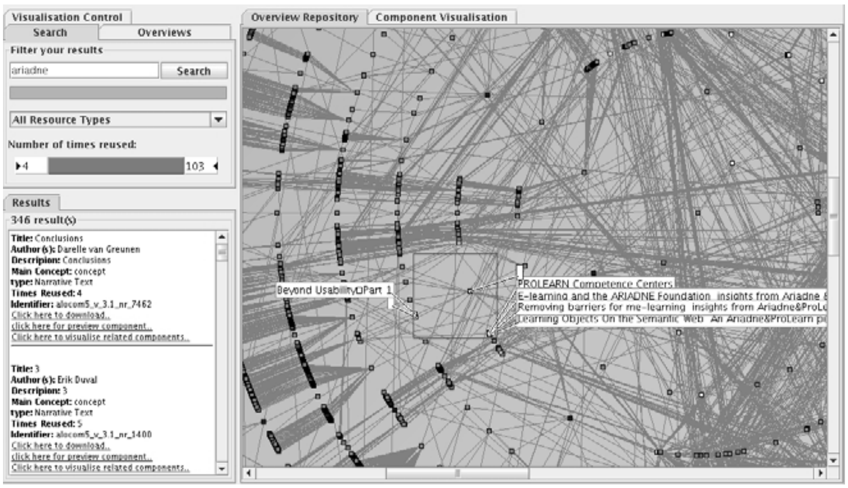


Figure 2: A screenshot of the visualization application of the ALOCOM Repository

3.1 User Interface

Figure 2 is a screenshot of the visualization application. The user interface consists of three components:

- The right panel uses more space than the other components and visualizes all the components in the ALOCOM repository.
- The left top part includes a number of filters that can be manipulated by the end user to filter out components that are not of interest.
- The left bottom part displays metadata on the relevant components.

In the following paragraphs, we will go in more detail on each of these parts and explain how this all works together.

3.1.1 The graph visualization

The visualization of the components in the ALOCOM repository consists of a graph with a twopi drawing layout [Wills, 97]. This algorithm tries to arrange all elements of the graph in circular or radial arcs, as can be clearly seen in figure 3. The vertices in the graph represent the different components in the repository. Colours are used to represent the granularity of the components. The edges represent a 'has part/is part of' relation between the different components in the repository.

The visualization helps the user to obtain an overview of reuse in the repository. As an example, figure 3 includes a number of separate components at the sides of the visualization. This means that there is no or little reuse of those components, compared to the other components in the repository. In the center of the figure, a big circle with many components displays substantial reuse between the different components in this circle. A more specific example is a figure in the centre of this circle about the Globe Consortium [GLOBE, 04]: this figure is reused in 6 different slides by different authors. Another example is a text-fragment about the Simple Query Interface [Simon, et al., 05] that is reused 110 times in 25 different slideshows. When one searches for a component in the repository by using the filter criteria and the results of this selection are found in a big circle with many components, one can immediately suspect that it is already reused a number of times.

Users of the application can interact directly with the visualization in a number of ways. First of all, they can pan through the visualization by scrolling through it and by using the overview-control, This shows a zoomed-out view of the visualization and a rectangle indicating the part that is currently visible. This rectangle can be moved to pan the visualization. Secondly the user can use excentric labeling. This is a technique where the user can label a neighbourhood of components that are located around the cursor [Fekete, et al., 99]. Finally, clicking on specific vertices in the graph visualization will lead the system to visualize only the clicked vertex and its related components. This new visualization will have less clutter, so that it will be easier to get insight in the reuse of the clicked component in other components and in the composition of the clicked component itself.

Indirectly, the user can interact with the visualization by using the filter controls that are described in the next section. The filters control which elements the visualization will fade out.

The graph drawing itself is performed by using the INRIA's Infovis Toolkit [Fekete, 04], an interactive open source toolkit, written in java, which aims to ease the development of visualization applications and components. In this toolkit, a software bridge in Java was implemented to the GraphViz [Ellson, et al., 03] toolkit, which essentially is a set of graph drawing algorithms and programs written in C. By plugging the infovis toolkit in our framework, we are automatically able to reuse this bridge and therefore all the graph drawing algorithms that are available in the GraphVis toolkit without having to create our own bridge. On top of that the infovis toolkit has a number of interesting information visualization techniques to interact with the visualization like e.g the use of fisheyes [Furnas, 99], magical lenses [Fishkin, M.C., 95], excentric labeling [Fekete, et al., 99], etc. which we can also reuse in our applications.

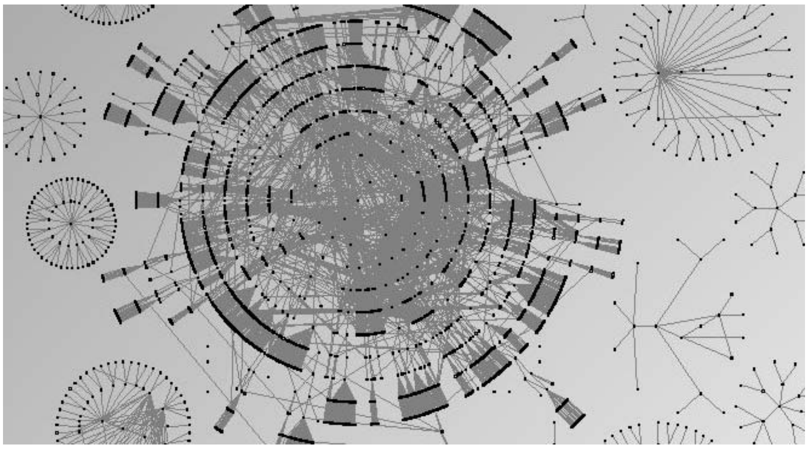


Figure 3: A graph with a radial tree that represents the ALOCOM Repository

3.1.2 The filter controls

With the filter controls, one can find different components of interest in an interactive and flexible way. There are a number of filter methods available to the user. With these methods, the user can:

- enter keywords.
- select resource types from a list that includes slides, slide, diagram, figure, table and narrative text.
- select an interval that represents a number of times that a component was reused in the repository.

By combining these filter methods, an end user can easily find figures about trees that are 5 times reused. The filter methods all work together with an AND-operator. Components that do not match the filter criteria are faded out in the visualization. Components that do match are highlighted. Moreover, metadata on the matching components are shown in the results information part of the interface.

3.1.3 The result information part

In this part, metadata is shown about the components that match the filter criteria. The user is able to select a result and either show it in the visualization, see a complete metadata instance about it, see a preview of the selected component, download and re-use it. For text fragment components, the text is shown in the preview; for slides, the slide is shown, etc.

4 Discussion and future work

Rapid-prototype developing allows the developers of an application to hold examples of their concepts for early visualization, verification, iteration, and optimisation. Our application was developed following this principle and we used the early development of the application to conduct some evaluation sessions. From these sessions we learned quite a number of interesting items.

First of all, the visualization of a repository of fine-grained components is a promising alternative to get insight in the components and their inner relations. Secondly, not only insight in the reuse of components can be obtained from the visualization, but also possible similar regions of interests between different authors that do not know each other can be obtained here when for instance an author notices that he or she uses the same sources of components as another author. With this information an overlapping social network could also be visualized, based on the reuse of different components.

Thirdly, we should ask authors of uploaded presentations to assess the automatic decomposition of their learning objects. Results of this study might indicate to what extent the automatic decomposition approaches a decomposition that is performed manually. Furthermore we plan to upload at least 1000 and preferably much more presentations to the ALOCOM repository. Based on the data we have currently available, this will result in approximately 50.000 components in the repository. This data set will be used to:

- calculate statistics about the reuse in learning objects in general.
- validate if the graph visualization will not get too cluttered as this will reduce the insight that can be obtained from the visualization.
- validate if the system will still have a high performance enough to access the components in an efficient and flexible way. This and the previous item are issues that many graph visualizations have to cope with. Further research will hopefully reveal a way how to tackle this scalability problem.

On top of this all we should also constantly track how users use this tool to be able to get a grasp on usability issues but also to enable us to measure the real value of this application. For this we plan to integrate the application with a logging system based on attention.XML that is developed in our research-group.

5 Conclusions

This paper reported on an interactive visualization application that aims to visualize a large repository of small reusable content components that were created by disaggregating legacy content. The main goal of the visualization application is to give an overview of the content components in the repository, their reuse and how they are put together in aggregate content. Furthermore, the application offers access to components and therefore it enables the reuse of these components in aggregates of the users that uses the application. Early evaluations revealed enthusiasm for the tool and validated the main goals of the application. Further evaluations will be conducted in the future when the ALOCOM Repository contains much more components.

Acknowledgements

We gratefully acknowledge the financial support of the K.U. Leuven Research Fund, in the context of the BALO project on "Basic research on Learning Objects" and of the European Commission through the ProLearn Network of Excellence.

References

[Baumgartner, et al., 04] J. Baumgartner, K. Börner, N. J. Deckard, and Nihar Sheth. "An XML Toolkit for an Information Visualization Software Repository"; Poster Compendium, IEEE Information Visualization Conference, pp. 72-73, 2003.

[Bederson, et al., 04] Bederson, B. B., Grosjean, J., & Meyer.J. "Toolkit Design for Interactiver Structured Graphics"; IEEE Transactions on Software Engineering, 30 (8), pp. 535-546.

[Card, et al., 99] Stuart K. Card, et al. "Readings in Information Visualization: Using Vision to Think"; Morgan Kaufmann series in interactive technologies, ISBN 1-55860-533-9, 686 pp.

[Cardinaels, et al., 2005] Kris Cardinaels, Michael Meire, Erik Duval, "Automating Metadata Generation: the Simple Indexing Interface", *International World Wide Web Conference Committee*, WWW 2005, May 10-14, 2005, Chiba, Japan.

[Ellson, et al., 03] J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. "Graphvis and Dynagraph – Static and Dynamic Graph Drawing Tools"; AT&T Labs – Research, Florham Park NJ 07932, USA.

[Fekete, et al.,, 99] Fekete, J.-D., Plaisant, C. "Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization" Revised version appears in the Proc. CHI'99, ACM , New York, pp. 512-519.

[Fekete, 04] Fekete, J-D., "The InfoVis Toolkit"; Proc. InfoVis'04, IEEE Press, 2004, 167-174.

[Furnas, 99] G. W. Furnas. "The FISHEYE view: a new look at structured files." *Readings in information visualization: using vision to think (1999*), Morgan Kaufmann Publishers, Inc., pp 312 – 330, ISBN:1-55860-533-9.

[Fishkin, M.C., 95] K. Fishkin, M.C. Stone. "Enhanced Dynamic Queries via Movable Filters". *Proceedings of CHI'95*, 1995, pp 415-420.

[GLOBE, 04] Global Learnng Objects Brokered Exchange: http://globe.edna.edu.au/globe/go

[Klerkx, et al., 04] J. Klerkx, E. Duval, and M. Meire. "Using information visualization for accessing learning object repositories"; Proc. IV04, IEEE, London, U.K (2004), 465-470.

[Klerkx, et al., 05] J. Klerkx, M. Meire, S.Ternier, K. Verbert, and E. Duval, "Information Visualization: Towards an Extensible Framework for Accessing Learning Object Repositories"; Proc. ED-Media'05, AACE, Montreal, Canada (2005), 4281-4287.

[Simon, et al., 05] Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., Miklós, Z. "A Simple Query Interface for Interoperable Learning Repositories." Workshop on Interoperability of Web-Based Educational Systems in conjunction with 14th International World Wide Web Conference (WWW'05). May, 2005, Chiba, Japan

[Ternier & Duval, 2003] S. Ternier, and E. Duval, Web services for the ARIADNE knowledge pool system, Proceedings of the 3rd Annual ARIADNE Conference (Duval, E., ed.), pp. 1-9, 2003.

[Verbert, et al., 2005] Verbert, K. et al, "Repurposing Learning Object Components", In Proc. of OTM 2005 Workshop on Ontologies, Semantics and E-learning, Agia Napa, Cyprus, 2005.

[Wills, 97] G.Wills. "interactive visualization of very large graphs." Symposium on Graph Drawing GD'97, volume 1353 of Lecture Notes in Computer Science, pages 403–414, 1997.