MindXpres: An Extensible Content-Driven Cross-Media Presentation Platform

Reinout Roels and Beat Signer

Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels, Belgium
{rroels,bsigner}@vub.ac.be

Abstract. Existing presentation tools and document formats show a number of shortcomings in terms of the management, visualisation and navigation of rich cross-media content. While slideware was originally designed for the production of physical transparencies, there is an increasing need for richer and more interactive media types. We investigate innovative forms of organising, visualising and navigating presentations. This includes the introduction of a new document format supporting the integration or transclusion of content from different presentations and cross-media sources as well as the non-linear navigation of presentations. We present MindXpres, a web technology-based extensible platform for content-driven cross-media presentations. The modular architecture and plug-in mechanism of MindXpres enable the reuse or integration of new visualisation and interaction components. Our MindXpres prototype forms a platform for the exploration and rapid prototyping of innovative concepts for presentation tools. Its support for multi-device user interfaces further enables an active participation of the audience which should ultimately result in more dynamic, engaging presentations and improved knowledge transfer.

Keywords: MindXpres, slideware, cross-media transclusion, non-linear navigation.

1 Introduction

With millions of digital presentations that are created every day, their importance in modern society cannot be denied. Presentations are supporting the oral transfer of knowledge and play an important role in educational settings. While digital presentation solutions originated as tools for creating physical media such as photographic slides or transparencies for overhead projectors, very little has changed to the underlying concepts and principles of slide-based presentation tools or so-called slideware for digital presentations. We are, for example, still restricted by the rectangular boundaries of a slide or the linear navigation between slides. As argued by Tufte [1], this form of slideware has some negative consequences for the effectiveness of knowledge transfer. Complex ideas are rarely sequential by nature but the presenter is forced to squeeze them into a linear sequence of slides, resulting in a loss of relations, overview and details. One might argue that these issues can easily be addressed by creating minimalistic presentations or by introducing some structure via a table of contents. Unfortunately, this does not work in the domain of learning, where complex knowledge or other pieces of rich information need to be presented "as is" [2].

It is important to point out the monolithic nature of slideware presentations where content is spread over many self-contained presentation files. In order to "reuse" previous work, the presenter has to switch between files while giving a presentation or duplicate some slides in the new presentation. Note that the issue is not limited to reusing single slides since there is a wealth of resources available, spread over a wide spectrum of distribution channels and formats. The inclusion of content by reference or transclusion [3] might help to cross the boundaries between different types of media and be beneficial in the context of modern cross-media presentation tools.

There also seems to be an imbalance between the functionality for the authoring and visualisation of content. The main authoring views consist of toolbars and buttons to specify how content should be visualised while there is less support for the authoring of the content itself. While we have seen the addition of basic multimedia types such as videos to modern slideware, most content is still rather static. During a presentation we can, for example, not easily change from a bar chart to a pie chart data visualisation or dynamically change some values to see the immediate effect, which could be beneficial for knowledge transfer [4]. Finally, the audience can be more actively involved via audience response and classroom connectivity systems which provide multi-device interfaces for sharing knowledge and results during as well as after a presentation. The evolution of presentations can be compared with the Web 2.0 movements where users have become contributors, content is more dynamic and interactive and where we have a decentralisation of content via service-oriented architectures.

The rapid prototyping and evaluation of new concepts for the representation, visualisation and interaction with content is essential in order to move a step towards the next generation of cross-media presentation tools. After introducing existing slideware solutions, we discuss the requirements for next generation presentation tools. This is followed by a description of the extensible MindXpres architecture and its plug-in mechanism. The web technology-based implementation of MindXpres is validated based on a number of use cases and MindXpres plug-ins and followed by a discussion of future work.

2 Background

The impact, benefits and issues of slideware have been studied ever since digital slideware has been introduced. While some studies acknowledge slideware as a teaching aid [4], Tufte [1] heavily criticises slideware for sticking to outdated concepts. He addresses the many consequences of spatial limitations or linear navigation and relates them to the human mind which works differently. One of Tufte's conclusions, which is also confirmed by Adams [5], is that slide-based presentations are not suitable for all kinds of knowledge transfer and in particular not in scientific settings. Recent work shows that it is important for the learning process that content is well integrated in the greater whole, both structurally and visually [6], which is influenced by the navigation and visualisation. There have been a number of different approaches to offer non-linear navigation. Zoomable User Interfaces (ZUIs) as used by CounterPoint [7], Fly [8] or Prezi, offer virtually unlimited space. Also Microsoft has experimented with zoomable interfaces in pptPlex. While ZUIs are one way to escape the boundaries of the slide, we have seen other approaches such as MultiPresenter [9] or tiling slideshows [10]. PaperPoint [11] and Palette [12] further enable the non-linear navigation of digital presentations based on a slide selection via augmented paper-based interfaces. Finally, there is a category of authoring tools that use hypermedia to enable different paths through a set of slides. NextSlidePlease [13] allows users to create a weighted graph of slides and may suggest navigational paths based on the link weights and the remaining presentation time. Microsoft follows this trend with their HyperSlides [14] project. The potential of PowerPoint as an authoring tool for hypermedia-based presentations has further been investigated by Garcia [15].

Existing presentation tools require content to be duplicated for reuse, resulting in multiple redundant copies that need to be kept up to date. Even though some attempts have been made to address this issue, there is room for improvement. When it comes to document formats for more general educational purposes, there are formats such as the Learning Material Markup Language (LMML) [16], the Connexions Markup Language (CNXML) and the eLesson Markup Language (eLML) [17]. The common factor of these formats is their focus on the reuse of content, but always at a relatively high granularity level. Content is organised in lessons or modules and users are encouraged to use these, as a whole, in their teaching. When we examined the formats in more detail, we noticed that they support outgoing links to external content but not the inclusion of content via references (transclusion). In the context of presentations, Microsoft's Slide Libraries are central repositories for the storage of slides in order to facilitate slide sharing and reuse within a company. However, one needs to set up a SharePoint server which might represent a hurdle for some users. Slides still need to be searched and manually copied into presentations. Furthermore, users are responsible to push back updates to the repository or update slides when they have been modified on the server side. Other commercial tools with similar intentions and functionality for content reuse are SlideRocket or SlideShare. The SliDL [18] research framework provides a service-oriented architecture for storing and tagging slides in a database for reuse. However, it shares some of the shortcomings of Microsoft's Slide Libraries. The ALOCOM [19] framework for flexible content reuse consists of a content ontology and a (de)composition framework for legacy documents including PowerPoint documents, Wikipedia pages and SCORM content packages. While ALOCOM succeeds in the decomposition of legacy documents, it might be too rigid for evolving presentation formats and the tool is furthermore only supporting the authoring phase.

There is not only a similarity in the evolution of the Web and presentation environments, but a number of the issues presented in this section have solutions in the setting of the Web. It is therefore not surprising that more recently we see the use of web technologies for realising presentation solutions. The Simple Standards-based Slide Show System (S5)[1] is an XHTML-based file format for slideshows which enforces the classical slideware model. The Slidy [20] initiative by the W3C introduces another presentation format which is based on the standard slideware model. While these two formats are too limited for our needs, they have some interesting properties. Both formats show a clean separation of content and presentation via CSS themes. The visualisation is resolution independent and the layout and font size are adapted to the available screen real estate. Finally, we would like to mention recent HTML5-based presentation solutions, including projects such as impress.js, deck.js, Shower or reveal.js. A major benefit of applying a widely used open standard such as HTML is the cross-device support. Nevertheless, also these solutions show some restrictions in terms of visualisation, navigation and cross-media support.

Most of the tools and projects presented in this section focus on specific novel ideas for presentations. However, there is no interoperability between the concepts introduced in different tools. While one project might focus on the authoring, another one focusses on novel content types and a third solution introduces radically new navigation mechanisms. Some slideware tools can be extended via third-party plug-ins but the functionality that is exposed to the developers is often limited by the tool's underlying model. For example, PowerPoint allows plug-ins to interact with the presentation model, but the model dictates that a presentation must consist of a sequence of slides. This lack of freedom is also a shortcoming of existing web-based presentation formats. We therefore see a need for an open presentation platform such as MindXpres which supports innovation by providing the necessary modularity and interoperability [21].

3 Requirements

We now introduce a number of requirements to support a broad range of presentation styles and visualisations which have been compiled based on a review of the more recent presentation solutions presented in the previous section.

Non-Linear Navigation. As outlined earlier, the linear traversal of slides is a concept that has been taken over from early photographic slides. Nowadays, users are accustomed to this form of navigation even if it might come with some disadvantages. Any navigation outside of the predefined linear path (e.g. to answer a question from the audience) is rather complicated, since the presenter either needs substantial time to scroll forwards or backwards to the desired slide or has to switch to the slide sorter view. It is further impossible to include a single slide multiple times in the navigational path without any duplication. There are different ways how this lack of flexible navigation can be addressed, including presentation tools that allow the presenter to define non-linear navigation paths [13,14] or zoomable user interfaces (ZUIs) [7,8,22].

[1] `http://meyerweb.com/eric/tools/s5/`

Separation of Content and Presentation. In order to facilitate experimentation with different visualisations, there should be a clear separation between content and presentation. This allows the authors of a presentation to focus on the content while the visualisation is handled by the presentation tool. Note that this approach is similar to the LaTeX typesetting system where content is written in a standardised structured way and the visualisation is automatically handled by the typesetting system. There is also a LaTeX document class for presentations called Beamer and we were inspired by its structured and content-driven approach. However, the content-related functionality and the visualisation are too limited to be considered as a basis for an extensible presentation tool.

Extensibility. In order for a presentation tool to be successful as an experimental platform for new presentation concepts, it should be easy to rapidly prototype new content types and presentation formats as well as innovative navigation and visualisation techniques. It has to be possible to add or replace specific components without requiring changes in the core. In order to be truly extensible, a presentation tool should provide a modular architecture with loosely coupled components. Note that this type of extensibility should not only be offered on the level of content types but also for the visualisation engine or content structures.

Cross-Media Content Reuse. In the introduction we briefly mentioned the lack of content reuse in existing presentation tools. There is a wealth of open education material available but it is rather difficult to use this content in presentations. On the other hand, the concept of transclusion works well for digital documents and parts of the Web (e.g. via the HTML `img` tag). A modern presentation tool should also support the seamless integration of external cross-media content. This includes various mechanisms for including parts of other presentations (e.g. slides), transcluding content from third-party document formats as well as including content from open learning repositories.

Connectivity. With the rise of social and mobile technologies, connectivity for multi-device input and output becomes more relevant in the context of presentation tools. Support for multi-directional connectivity is required for a number of reasons. First, it is necessary for the previously mentioned cross-media transclusion from external resources. Second, multi-directional connectivity forms the backbone for audience feedback via real-time response or voting systems [23] as well as other forms of multi-device interfaces.

Interactivity. We mentioned that content might be more interactive and the extensibility requirement addresses this issue since the targeted architecture should support dynamic or interactive content and visualisations. Nevertheless, the use of mouse and keyboard might not be sufficient for components offering a high level of interaction. Therefore, a presentation tool should enable the integration of other forms of input such a gesture-based interaction based on Microsoft's Kinect controller or digital pen interaction [24] as offered by the PaperPoint [11] presentation tool.

Post-Presentation Phase. Even if it was never the original goal of slide decks, they often play an important role as study or reference material. While the sharing of traditional slide decks after a presentation is trivial, this changes when the previously mentioned requirements are taken into account. For instance, the non-linear navigation allows presenters to go through their content in a non-obvious order or input from the audience might drive parts of a presentation. Special attention should therefore be paid to the post-presentation phase. It should not only be easy to play back a presentation with the original navigational path, annotations and audience input, but its content should also be made discoverable and reusable. In accordance with the Web 2.0, we see potential for the social aspect in a post-presentation phase via a content discussion mechanism.

4 MindXpres Platform

In this section, we present the general architecture of our MindXpres[2] cross-media presentation platform which is outlined in Fig. 1 and addresses the requirements presented in the previous section.

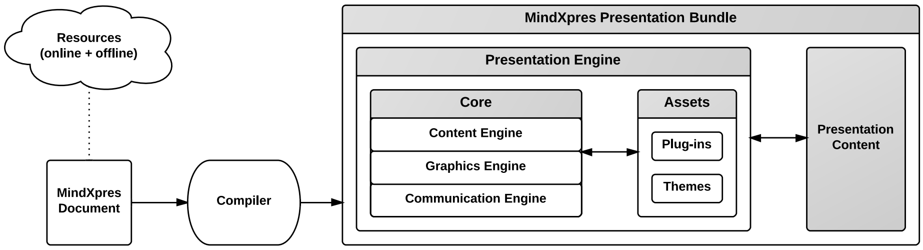


Fig. 1. MindXpres architecture

4.1 Document Format and Authoring Language

Content is stored, structured and referenced in a dedicated MindXpres document format. An individual MindXpres document contains the content itself and may also refer to some external content to be included. A new MindXpres document can be written manually similar to the LaTeX approach introduced earlier or in the near future it can also be generated via a graphical authoring tool. In contrast to other presentation formats such as Slidy, S5 or OOXML, the authoring language eliminates unnecessary HTML and XML specifics and focusses on a semantically more meaningful vocabulary. The vocabulary of the authoring language is almost completely defined by plug-ins that provide support for various media types and structures. In order to give users some freedom in the way they present their information, the core MindXpres presentation engine only plays a

[2] `http://mindxpres.com`

supporting role for plug-ins and lets them define the media types (e.g. video or source code) as well as structures (e.g. slides or graph-based content layouts). This is also reflected in the document format as each plug-in extends the vocabulary that can be used. Any visual styling including different fonts, colours or backgrounds is achieved by applying specific themes to the underlying content.

4.2 Compiler

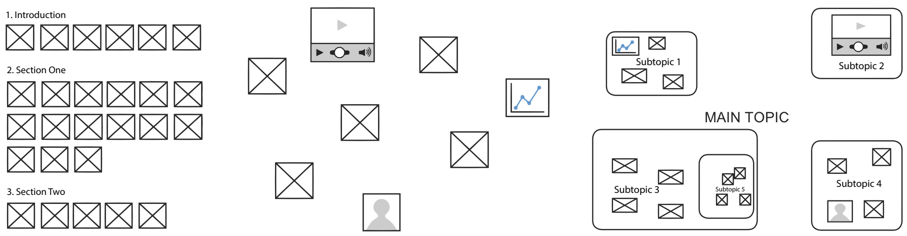
The compiler transforms a MindXpres document into a self-contained portable MindXpres presentation bundle. While a MindXpres document could be directly interpreted at visualisation time, for a number of reasons we decided to have this intermediary step. First, the compiler allows different types of presentations to be created from the same MindXpres document instance. This means that we can not only create dynamic and interactive presentations but also more static output formats such as PDF documents for printing. Similarly, we cannot always expect that there will be an Internet connection while giving a presentation. For this case, the compiler might create an offline version of a presentation with all necessary content pre-downloaded and included in the MindXpres presentation bundle. Last but not least, the compiler might resolve incompatibility issues by, for instance, converting unsupported video formats.

4.3 MindXpres Presentation Bundle

The dynamic MindXpres presentation bundle consists of the compiled content together with a portable cross-platform presentation runtime engine which allows more interactive and networked presentations. Similar to the original document, the compiled presentation content still consists of both, integrated content and references to external resources such as online content that will be retrieved when the presentation is visualised. Note that the content might have been modified by the compiler and, for example, been converted or extracted from other document formats that the runtime engine cannot process. References to external content may have been dereferenced by the compiler for offline viewing.

A presentation bundle's core runtime engine consists of the three modules shown in Fig. 1. The *content engine* is responsible for processing the content and linking it to the corresponding visualisation plug-ins. The *graphics engine* abstracts all rendering-related functionality. For instance, certain presenters prefer a zoomable user interface in order to provide a better overview of their content [25]. This graphical functionality is also exposed to the plug-ins, which can make use of the provided abstractions. The *communication engine* exposes a communication API that can be used by plug-ins. It provides some basic functionality for fetching external content but also offers the possibility to form networks between multiple MindXpres presentation instances as well as to connect to third-party hardware such as digital pens or clicker systems.

In addition to the presentation content and core modules, the presentation bundle contains a set of *themes* and *plug-ins* that are referenced by the content.



(a) Structured **(b)** Graph **(c)** Clustered

Fig. 2. Structure plug-in examples

Themes may contain visual styling on a global as well as on a plug-in level. When the content engine encounters different content types, they are handed over to the specific plug-in which uses the graphics engine to visualise the content.

4.4 Plug-in Types

In order to provide the necessary flexibility, all non-core modules are implemented as plug-ins. Even the basic content types such as text, images or bullet lists have been realised via plug-ins and three major categories of plug-ins have to be distinguished:

– *Components* form the basic building blocks of a presentation. They are represented by plug-ins that handle the visualisation for specific content types such as text, images, bullet lists, graphs or videos. The content engine invokes the corresponding plug-ins in order to visualise the content.
– *Containers* are responsible for grouping and organising components of a specific type. An example of such a container is a slide with each slide containing different content but also some reoccurring elements. Every slide of a presentation may for example contain elements such as a title, a slide number and the author's name, which can be abstracted in a higher level container. Another example is an image container that visualises its content as a horizontally scrollable list of images. Note that we are not restricted to the slide format and content can be laid out in alternative ways.
– *Structures* are high-level structures and layouts for components and containers. For example, content can be scattered in a graph-like structure or it can be clearly grouped in sections like in a book. Both are radically different ways of visualising and navigating content but by abstracting them as plug-ins, the user can easily switch between different presentation styles as the ones shown in Fig. 2. Structures differ from containers by the fact that they do not impose restrictions on the media types of their child elements and may also influence the default navigational path through the content.

5 Implementation

We have chosen HTML5 and its related web technologies as the backbone for our MindXpres presentation platform. Other options such as JavaFX, Flash or game engines have also been investigated, but HTML5 seemed to be the best choice. The widely accepted HTML5 standard makes MindXpres presentations highly portable and runnable on any device with a recent web browser, including smartphones and tablets. Furthermore, HTML5 provides rich visualisation functionality out of the box and the combination with Cascading Style Sheets (CSS) and third-party JavaScript libraries forms a potent visualisation platform.

5.1 Document Format and Authoring Language

The MindXpres document format which allows us to easily express a presentation's content, structure and references is based on the eXtensible Markup Language (XML). A simple example of a presentation defined in our XML-based authoring language is shown in List. 1.1. The set of valid tags and their structure, except the `presentation` root tag, is defined by the available plug-ins.

```xml
<presentation>
  <slide title="Vannevar Bush">
    <bulletlist>
      <item>March 11, 1890 - June 28, 1974</item>
      <item>Amercian Engineer, founder of Raytheon</item>
    </bulletlist>
    <image source="bush.jpg"/>
  </slide>
</presentation>
```

Listing 1.1. Authoring a simple MindXpres presentation

5.2 Compiler

The compiler has been realised as a Node.js application. This not only allows the compiler to be used via a web interface or as a web service, but projects such as node-webkit also enable the compiler to run as a local offline desktop application. The choice of using server-side JavaScript was influenced by the fact that Node.js is capable of bridging web and desktop technologies. On the one hand, the framework makes it easy to interact with other web services and to work with HTML, JSON, XML and JavaScript visualisation libraries at compile time. On the other hand, the framework can also perform tasks which are usually not suited for web technologies, including video conversion, legacy document format access, file system access or TCP/IP connectivity.

In order to validate a MindXpres document in the XML format described above, there is an XML Schema which is augmented with additional constraints provided by the plug-ins. After validation, the document is parsed and discovered tags might trigger preprocessor actions by the plug-ins such as the extraction of data from referenced legacy document formats (e.g. PowerPoint or Excel) or the conversion of an unsupported video format. The tag is then converted to HTML5 by simply encoding the information in the attributes of a `div` element. The HTML5 standard allows custom attributes if they start with a `data-` prefix. Listing 1.2 shows parts of the transformed XML document shown in List. 1.1. Note that the transformation does not include visualisation-specific information but merely results in a valid HTML5 document which is bundled into a self-contained package together with the presentation engine.

```
<div data-type="presentation">
  <div data-type="slide" data-title="Vannevar Bush">
    <div data-type="bulletlist">
    ...
```

Listing 1.2. Transformed HTML5 presentation content

5.3 Presentation Engine

The presentation engine's task is to turn the compiled HTML content into a visually appealing and interactive presentation. As highlighted in Fig. 1, the presentation engine consists of several smaller components which help plug-ins to implement powerful features with minimal effort. The combination of these components enables the rapid prototyping and evaluation of innovative visualisation ideas. A resulting MindXpres presentation combining various structure, container and component plug-ins is shown in Fig. 3.

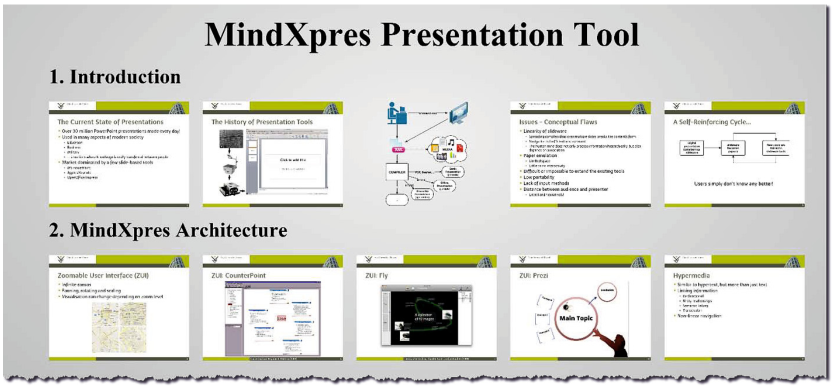

MindXpres Presentation Tool
1. Introduction
The Current State of Presentations
The History of Presentation Tools
Issues – Conceptual Flaws
A Self-Reinforcing Cycle...
Users simply don't know any better!
2. MindXpres Architecture
Zoomable User Interface (ZUI)
ZUI: CounterPoint
ZUI: Fly
ZUI: Prezi
Main Topic
Hypermedia

Fig. 3. A MindXpres presentation

Content Engine. When a presentation is loaded, the content engine is the first component that is activated. It processes the content of the HTML presentation by making use of the well-known jQuery JavaScript library. Whenever a `div` element is discovered, the `data-type` attribute is read and the corresponding plug-ins are notified in order to visualise the content.

Graphics Engine. The graphics engine provides support for interesting new visualisation and navigation styles. Next to some basic helper functions, it offers efficient panning, scaling and rotation via CCS3 transformations and supports zoomable user interfaces as well as the more traditional navigation approaches.

Communication Engine. The communication engine implements abstractions that allow plug-ins to retrieve external content at run time. It further provides the architectural foundation to form networks between different MindXpres instances or to integrate third-party hardware [26]. For our MindXpres prototype, we used a small Intel Next Unit of Computing Kit (NUC) with high-end WiFi and Bluetooth modules to act as a central access point and provide the underlying network support. MindXpres instances use WebSockets to communicate with other MindXpres instances via the access point. The access point further acts as a container for data adapters which translate input from third-party input and output devices into a generic representation that can be used by the MindXpres instances in the network. In order to go beyond simple broadcast-based communication, we have implemented a routing mechanism based on the publish-subscribe pattern where plug-ins can subscribe to specific events or publish information. The communication engine provides the basis for audience response systems [26] or even full classroom communication systems where functionality is only limited by the creativity of plug-in developers.

Plug-ins. Plug-ins are implemented as JavaScript bundles which consist of a folder containing JavaScript files and other resources such as CSS files, images or other JavaScript libraries. As a first convention, a plug-in should provide a manifest file with a predefined name. The manifest provides metadata such as the plug-in name and version but also a list of tags to be used in a presentation. The plug-in claims unique ownership for these tags and is in charge for their visualisation if they are encountered by the content engine. As a second convention, a plug-in must provide at least one JavaScript file implementing certain methods, one of them being the `init()` method which is called when the plug-in is loaded by the presentation engine. It is up to the plug-in to load additional JavaScript or CSS via the provided dependency injection functionality. A second method to be implemented is the `visualise()` method which is invoked with a pointer to the corresponding DOM node as a parameter when the content engine encounters a tag to be visualised. A plug-in is free to modify the DOM tree and may also register callbacks to handle future interaction with the content.

Themes. We currently use CSS to provide a basic templating system. These themes offer styling either on a global or on a plug-in level. However, we see this as a temporary solution as it is not well-suited for alternative compiler outputs (e.g. PDF) and a more generic templating scheme is planned for the future.

6 Use Cases

In order to validate the architectural and technological choices, we demonstrate the extensibility and feasibility of MindXpres as a rapid prototyping platform by presenting a number of content- and navigation-specific plug-ins that have been developed so far. Additional plug-ins for audience-driven functionality such as real-time polls, screen mirroring and navigational takeover can be found in [26].

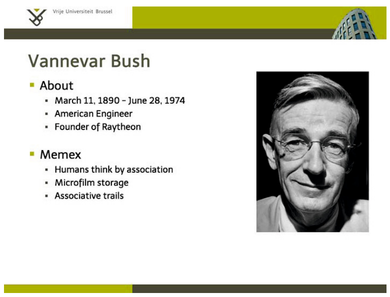
6.1 Structured Overview Plug-in

In Sect. 4 we have explained how structure plug-ins may change the way presentations are visualised and navigated. In order to illustrate this, we have implemented a structure plug-in called *structured layout* which combines a zoomable user interface with the ability to group content into sections. The resulting visualisation of the *structured layout* plug-in is shown in Fig. 3. Whenever a new section is reached, the view is zoomed out to provide an overview of the content within the section and communicate a sense of progress.

6.2 Slide Plug-in

In order to also support the traditional slide concept, we created a slide-like container plug-in. While the benefits and issues of using slides with a fixed size are debatable, we implemented this plug-in as a proof of the framework's versatility. The main function of the slide plug-in is to provide a rectangular styleable component container with a title and some other information. Containers may also offer functionality to layout their content. In this case, the slide plug-in offers a quick and easy layout mechanism which allows the presenter to partition the slide into rows and columns. Content is then assigned to these slots in the order that it is discovered. The use of the slide plug-in together with the resulting visualisation is exemplified in Fig. 4. It also shows the use of the image plug-in

```
<slide layout="\60\40" title="Vannevar Bush">
 <bulletlist>
  <item>About
   <item>March 11, 1890 - June 28, 1974</item
        >
   <item>American Engineer</item>
   <item>Founder of Raytheon</item>
  </item>
  <item>...
  </item>
 </bulletlist>
 <image source="http://example.com/bush.jpg">
  <crop bounds="10%, 5%, -10%, -20%" />
 </image>
</slide>
```



Vannevar Bush

- About
  - March 11, 1890 – June 28, 1974
  - American Engineer
  - Founder of Raytheon
- Memex
  - Humans think by association
  - Microfilm storage
  - Associative trails

Fig. 4. Slide plug-in

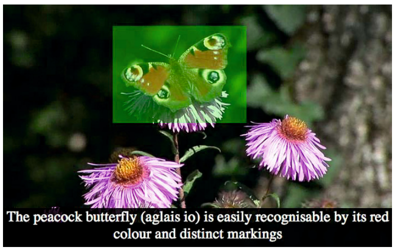
(a component plug-in) which enables a simple form of cross-media transclusion. A visualised external image can be cropped and filters (e.g. colour correction) may be applied without duplicating or modifying the original source.

6.3 Enhanced Video Plug-in

When videos are used in educational settings, we often need more functionality than what is offered by the average video player [25]. MindXpres provides the enhanced video plug-in shown in Fig. 5 with the possibility to overlay a video with text or arbitrary shapes. This overlay functionality can be used as a basic captioning system as well as to highlight items of interest during playback.

Furthermore, we added the option to trigger certain events at specified times. One can define that a video should automatically pause at a certain point, highlight an object and continue playing after a specified amount of time. Additional features include the bookmarking of certain positions in a video for direct access or the possibility to display multiple videos in a synchronised manner. Our enhanced video plug-in injects the default HTML5 video player and overlays it with a transparent `div` element for augmentation. Currently we make use of the HTML5 video API to synchronise the creation and removal of overlays but a SMIL-based implementation might be used in the future.

```
<video source="vid.mp4">
  <caption start="0:00" duration="1500ms">
    Lecture 3 - Butterfly Species
  </caption>
  <pause start="0:43" duration="5s">
    <caption>
      The peacock butterfly (aglais io) ...
    </caption>
    <highlight x="30%" y="9%"
               width="35%" height="40%" />
  </pause>
</video>
```



The peacock butterfly (aglais io) is easily recognisable by its red colour and distinct markings

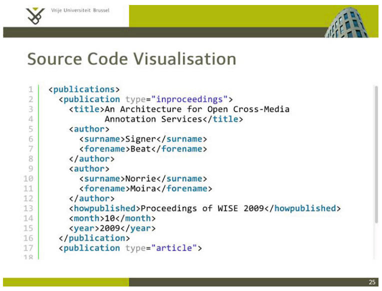
Fig. 5. Enhanced video plug-in

6.4 Source Code Visualisation Plug-in

Earlier, we mentioned the difficulty of visualising complex resources such as source code. Our MindXpres source code plug-in exports a `code` tag which allows the presenter to paste their code into a presentation and have MindXpres visualise it nicely by making use of syntax highlighting via the SyntaxHighlighter[3] JavaScript library. Whenever the content engine encounters a `code` tag, it invokes the code plug-in to beautify the code and automatically adds vertical scrollbars for larger pieces of source code as shown in Fig. 6.

[3] http://alexgorbatchev.com/SyntaxHighlighter/

```
<code>
 <publications>
  <publication type="inproceedings">
   <title>An Architecture for Open Cross-Media
          Annotation Services</title>
   <author>
    <surname>Signer</surname>
    <forename>Beat</forename>
   </author>
   <author>
    <surname>Norrie</surname>
    <forename>Moira</forename>
    ...
</code>
```



Source Code Visualisation

```
<publications>
  <publication type="inproceedings">
    <title>An Architecture for Open Cross-Media
           Annotation Services</title>
    <author>
      <surname>Signer</surname>
      <forename>Beat</forename>
    </author>
    <author>
      <surname>Norrie</surname>
      <forename>Moira</forename>
    </author>
    <howpublished>Proceedings of WISE 2009</howpublished>
    <month>10</month>
    <year>2009</year>
  </publication>
  <publication type="article">
```

Fig. 6. Source code visualisation

7 Discussion and Future Work

MindXpres currently supports transclusion and cross-media content reuse on the plug-in level. For instance, the image or video plug-in can visualise (and enhance) external resources, a dictionary plug-in might retrieve definitions on demand via a web service or we might create a plug-in that allows us to import content (e.g. PowerPoint slides) from legacy documents at compile time. Nevertheless, we are currently investigating the introduction of generic reuse tags in our document format which would allow the presenter to transclude arbitrary parts of other MindXpres presentations. While our focus has been on the cross-media aspect of resources that can be used in a presentation, we might also investigate the cross-media publishing aspect via alternative compiler output formats.

We are aware that the current authoring of MindXpres presentations has some usability issues. The average presenter cannot be expected to construct an XML document or any CSS themes. In order to tackle this issue and further evaluate MindXpres in real-life settings, we are currently developing a graphical MindXpres authoring tool. We further intend to provide a central plug-in repository which would make it easy for novice users to find, install and use new plug-ins via the authoring tool. In the long run, we intend to revise the use of monolithic documents and move towards repositories of semantically linked information based on the RSL hypermedia metamodel [27]. This would not only promote content reuse and sharing, but also create opportunities for context-aware as well as semi-automatic presentation authoring where relevant content is recommended by the authoring tool.

8 Conclusion

We have presented MindXpres, an HTML5 and web technology-based presentation platform addressing the lacking extensibility of existing slideware solutions. The extensibility of the MindXpres platform heavily relies on a plug-in mechanism and we have outlined how the tool can be extended on the content, visualisation as well as on the interaction level. By providing different forms of cross-media content transclusion, our solution further avoids the redundant storage and replication of slides. The presented MindXpres document format in combination with a compiler and the corresponding plug-ins offers the opportunity to have compile- or run-time cross-media content transclusion from third-party resources. At the same time, the flexible and extensible document model in combination with a zoomable user interface allows us to escape the boundaries of traditional slide formats. The presented MindXpres solution represents a promising platform for the unification of existing presentation concepts as well as for the rapid prototyping and investigation of new ideas for next generation cross-media presentation solutions.

References

1. Tufte, E.R.: The Cognitive Style of PowerPoint: Pitching Out Corrupts Within. Graphics Press (2003)
2. Farkas, D.K.: Toward a Better Understanding of PowerPoint Deck Design. Information Design Journal + Document Design 14(2) (August 2006)
3. Nelson, T.H.: The Heart of Connection: Hypermedia Unified by Transclusion. Communications of the ACM 38(8) (August 1995)
4. Holzinger, A., Kickmeier-Rust, M.D., Albert, D.: Dynamic Media in Computer Science Education; Content Complexity and Learning Performance: Is Less More? Educational Technology & Society 11(1) (January 2008)
5. Adams, C.: PowerPoint, Habits of Mind, and Classroom Culture. Journal of Curriculum Studies 38(4) (2006)
6. Gross, A., Harmon, J.: The Structure of PowerPoint Presentations: The Art of Grasping Things Whole. IEEE Transactions on Professional Communication 52(2) (December 2009)
7. Good, L., Bederson, B.B.: Zoomable User Interfaces as a Medium for Slide Show Presentations. Information Visualization 1(1) (March 2002)
8. Lichtschlag, L., Karrer, T., Borchers, J.: Fly: A Tool to Author Planar Presentations. In: Proc. of CHI 2009, Boston, USA (April 2009)
9. Lanir, J., Booth, K.S., Tang, A.: MultiPresenter: A Presentation System for (Very) Large Display Surfaces. In: Proc. of MM 2008, Vancouver, Canada (October 2008)
10. Chen, J.C., Chu, W.T., Kuo, J.H., Weng, C.Y., Wu, J.L.: Tiling Slideshow. In: Proc. of Multimedia 2006, Santa Barbara, USA (October 2006)
11. Signer, B., Norrie, M.C.: PaperPoint: A Paper-based Presentation and Interactive Paper Prototyping Tool. In: Proc. of TEI 2007, Baton Rouge, USA (February 2007)
12. Nelson, L., Ichimura, S., Pedersen, E.R., Adams, L.: Palette: A Paper Interface for Giving Presentations. In: Proc. of CHI 1999, Pittsburgh, USA (May 1999)
13. Spicer, R., Lin, Y.R., Kelliher, A., Sundaram, H.: NextSlidePlease: Authoring and Delivering Agile Multimedia Presentations. ACM TOMCCAP 8(4) (November 2012)
14. Edge, D., Savage, J., Yatani, K.: HyperSlides: Dynamic Presentation Prototyping. In: Proc. of CHI 2013, Paris, France (April 2013)
15. Garcia, P.: Retooling PowerPoint for Hypermedia Authoring. In: Proc. of SITE 2004, Atlanta, USA (March 2004)
16. Süß, C., Freitag, B.: LMML - The Learning Material Markup Language Framework. In: Proc. of ICL 2002, Villach, Austria (September 2002)

17. Fisler, J., Bleisch, S.: eLML, the eLesson Markup Language: Developing sustainable e-Learning Content Using an Open Source XML Framework. In: Proc. of WEBIST 2006, Setubal, Portugal (April 2006)
18. Canós, J.H., Marante, M.I., Llavador, M.: SliDL: A Slide Digital Library Supporting Content Reuse in Presentations. In: Lalmas, M., Jose, J., Rauber, A., Sebastiani, F., Frommholz, I. (eds.) ECDL 2010. LNCS, vol. 6273, pp. 453–456. Springer, Heidelberg (2010)
19. Verbert, K., Ochoa, X., Duval, E.: The ALOCOM Framework: Towards Scalable Content Reuse. Journal of Digital Information 9(1) (2008)
20. Raggett, D.: Slidy - A Web Based Alternative to Microsoft PowerPoint. In: Proc. of XTech 2006, Amsterdam, The Netherlands (May 2006)
21. Bush, M.D., Mott, J.D.: The Transformation of Learning with Technology: Learner-Centricity, Content and Tool Malleability, and Network Effects. Educational Technology Magazine 49(2), 3–20 (2009)
22. Haller, H., Abecker, A.: iMapping: A Zooming User Interface Approach for Personal and Semantic Knowledge Management. In: Proc. of Hypertext 2010, Toronto, Canada (June 2010)
23. Dufresne, R.J., Gerace, W.J., Leonard, W.J., Mestre, J.P., Wenk, L.: Classtalk: A Classroom Communication System for Active Learning. Journal of Computing in Higher Education 7(2) (1996)
24. Signer, B., Norrie, M.C.: Interactive Paper: Past, Present and Future. In: Proc. of PaperComp 2010, Copenhagen, Denmark (September 2010)
25. Reuss, E.I., Signer, B., Norrie, M.: PowerPoint Multimedia Presentations in Computer Science Education: What do Users Need? In: Holzinger, A. (ed.) USAB 2008. LNCS, vol. 5298, pp. 281–298. Springer, Heidelberg (2008)
26. Roels, R., Signer, B.: A Unified Communication Platform for Enriching and Enhancing Presentations with Active Learning Components. In: Proc. of ICALT 2014, Athens, Greece (July 2014)
27. Signer, B., Norrie, M.: As We May Link: A General Metamodel for Hypermedia Systems. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 359–374. Springer, Heidelberg (2007)