



Co-funded by the
Erasmus+ Programme
of the European Union



Features for VISION

– University of Amsterdam Team: Ivan de Wolf, Ana Varbanescu, Natasa Brouwer-Zupancic, Erwin van Vliet, Ana Oprescu

Introduction

This document presents the different [authoring tool](#) and [chatbot](#) features that comply with state-of-the-art systems and are in line with the requirements from VISION – Virtual Interface for Smart Interaction Online. This document is divided into two sections. The first section discusses the features of existing authoring tools while the second section goes into existing and potential features for a chatbot.

Authoring tool

To discuss the features required for the authoring tool, we analysed different existing state-of-the-art tools. For each feature, existing tools implementing that feature are also listed.

WYSIWYG interface The interface should be non-technical and easy to use. A what-you-see-is-what-you-get (WYSIWYG) interface will give direct feedback to the author on how the content will be displayed. [*Elucidat, Adobe Captivate, Storyline 360, Gomo, Lectora*]

Template interface There should be an interface where it is possible to build templates which can be reused. This should include some mechanism to copy the look-and-feel from one module to another, when, for example, a course/an author uses multiple modules. [*Elucidat, Adobe Captivate, Storyline 360, Gomo*]

Responsive content Students consume content on all kinds of different devices, including laptops, tablets and smartphones. It is essential that the content is accessible from every device. [*Elucidat, Adobe Captivate, Gomo, Lectora*]

Importing existing material On a lot of existing tools the importing of old material is not recommended, but this is an absolute must to implement, because there is a lot of existing material. Therefore, an essential feature is the ability to import existing materials which are mostly powerpoints and PDFs. *[Lectora]*

Intuitive interface A lot of feedback on the current Authoring Tools is about the steep learning curve the tools have for new users. Especially for teachers with little experience in digital courses. Thus, the interface should be intuitive to use, to reduce the steepness of the learning curve. *[Storyline 360]*

Co-authoring When authoring content, authors might want to collaborate. Therefore, especially for courses with multiple authors, a nice feature to have is to allow sharing of content with other authors. *[Elucidat, Storyline 360, Gomo]*

SCORM, HTML5, xAPI (TinCan), LTI When an author is done, it should be possible to output the content using different methods. The most used ones are SCORM, HTML5, xAPI (TinCan) and LTI. *[Elucidat, Adobe Captivate, Storyline 360, Gomo, Lectora]*

Cloud-based There are a lot of advantages to making a cloud-based authoring tool. Authors can use the tool on every platform and updating the tool automatically makes sure every user is on the latest version. *[Elucidat, Gomo]*

Media support To create a wide variety of content as an author, the tool must support a wide range of media types, including, but not limited to text, images, video and audio. Some tools are also starting to support the use of VR-content, and some even allow the author to directly record their screen from the application to easily add instructions. *[Adobe Captivate, Storyline 360, Lectora]*


Chatbot

Scenario: using IGuideMe with the VISION chatbot, use TicketVise to get questions in a first phase, and then use TicketVise as notification mechanism (and close feedback loop) for when the bot does not have an answer.

Scenario: exam simulation?

Chatbots are just starting to be used in education, which means there is a lot of potential to discover. Especially in large courses, with over a hundred enrolled students, chatbots can help with a more personalized learning environment. A list of desirable features, together with a list of existing chatbots implementing each feature, is presented below.

Course questions A basic beginning for a chatbot is to answer basic questions about information provided by the teacher (course manual or other sources), such as office hours and examination duration. *[Uji]*



Material directions The next step is to support the search for course material. It would help a student with a question like: "Where can I find *this or that?*" to respond with the directions to a specific PDF or Powerpoint with possibly the page or section number. More complex questions can also be envisioned, like searching for specific items, such as "definitions" or "exercises" or "common mistakes" related to content. **Note:** the granularity of the questions and material must match: I cannot ask about DRAM memory technology if the only "item" in the AT is "Computer architecture", without any subdivision called "Memory system".

Switch to human When the chatbot does not give the right answer a student will be redirected to a course teacher. The chatbot should be supporting this request and a teacher or assistant needs to get a notification in some way to respond to the question (enter a tool like TicketVise).

Course Integration Another important feature is where the chatbot 'lives' and how it is accessible for teachers and students. Preferably the chatbot should live as close to the course as possible. This way when a student has a question about the material they can directly access and ask the chatbot. *[Mongoose Harmony]*

Automating tasks In addition to answering questions from students, the chatbot can also automate a set of tasks. For example, the chatbot could scan an uploaded document for feedback. *[Mongoose Harmony, Amazon's QnABot]*

Personal tracking To make the chatbot more personal, it could track the progress of a student with the material and when the student has a question about the material the chatbot can offer different materials if the student has more questions and not keep responding with the same answer. This could also be used in combination with assignments and grades when a student asks on which topic he needs to focus for the next exam. *[IBM Watson, Juji]*

Proactive An advanced step would be to make the chatbot proactive and message students about their progress, point out certain topics that should be read again/understood better, or remind them they need to finish a set of administrative tasks. For example, sending reminders for official course deadlines of internal, own study-plans. This proactive behaviour will make use of external student analytics tools *[Juji]*

"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."