# *A Framework for the Delivery of Personalized Adaptive Content*

Colm Howlin
*CCKF Limited*
*Dublin, Ireland*
*colm.howlin@cckf-it.com*

Danny Lynch
*CCKF Limited*
*Dublin, Ireland*
*colm.howlin@cckf-it.com*

***Abstract*— The number of students taking online courses has grown dramatically in recent years. Increasingly institutions are turning to adaptive platforms to deliver these online courses as they seek to offer personalized learning to each individual. The goal is to allow learners to progress at their own pace, place and convenience. Key to any learning system is the content through which the learning is delivered. Within the approach used by Realize[it] learning platform, curriculum and content are separated. The curriculum is used to drive the direction of the personalized learning path, while the content is responsible for the delivery and presentation of the concepts to be learned. This paper outlines the content framework behind this approach which can enable the delivery of personalized adaptive learning.**

## I. Introduction

The number of students taking online courses has grown dramatically in recent years. As of Fall 2012 online enrollments in the United States has surpassed 7.1 million and now accounts for 33% of total higher education enrollments (21.3 million) [1]. Traditionally, most institutions used learning management systems to deliver these courses. These systems provided the same courses in the same structure, composition, and content for all students [2]. However each learner has their own set of needs and characteristics, such as prior knowledge and learning style, which can determine their outcomes on a course [3]. This forms the basis of the argument that instruction should be adapted and personalized to learners, [4]. Increasingly institutions are turning to adaptive platforms to not only take their courses online, but to exploit the differences between learners and to deliver personalized and adaptive learning to every student.

Modern online learning systems which are both adaptive and data driven have the potential to independently provide both a personalized learning experience for the student and to support instructors and administrators in creating, administering, and managing online courses. In addition to this, the vast wealth of data collected by such systems can allow all stakeholders, including the student, insights into the learning process.

For any learning system to be effective it ultimately relies upon its learning content. It is through the content that the majority of student interactions with the system take place and where the intelligence and adaptivity will be most visible. It is the content which is the focus of this paper. A framework will be described which can enable a data driven approach to the delivery of personalized and adaptive content, but first a short description of the approach of the Realize[it] platform, within which this frameworks has been implemented, will be given.

## II. The Adaptive Learning Platform

The fundamental approach behind this framework, and the platform in which it sits, is to separate the curriculum from the content. In any learning environment, placing a learner in a situation where they must both navigate a curriculum and select content places an enormous additional cognitive load on them. A teacher avoids this by making decisions and suggestions to the learners as they learn. This framework achieves this through the use of its Artificial Intelligence Engine (AIE) which is a collection of algorithms that help bridge the gap between the curriculum, content and the learner, *Fig 1*.

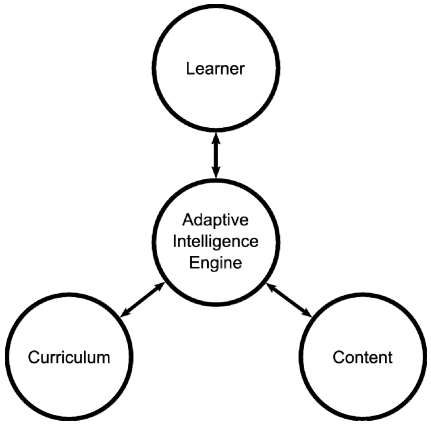


Fig 1. The separation of the curriculum from the content, and monitoring the interaction of both with the learner by the Adaptive Intelligence Engine

The interaction of the learner with both the curriculum and the content generates a rich stream of data which the engine must constantly analyze and mine to create a detailed and relevant model of each. The more relevant information in the models the better the adaptation and personalization, [6].

### *A. Curriculum*

Traditionally a curriculum is defined by a hierarchical structure with the knowledge items, the smallest piece of discrete knowledge, at the bottom of the structure. This

978-1-4799-5739-2/14/$31.00 ©2014 IEEE

framework supplements this structure with a second structure, known as the Curriculum Prerequisite Network (CPN). This is a directed acyclic graph where knowledge items are nodes and the edges represent the prerequisite relationships that exist between knowledge items, Fig 2.

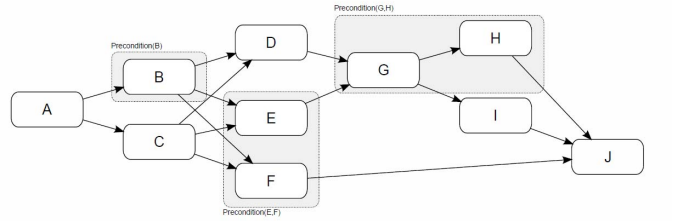


Fig 2. A sample Curriculum Prerequisite Network.

*Preconditions:* In addition to the prerequisite links in the CPN a single or set of knowledge items can have a list of preconditions that must also be fulfilled before they can be undertaken. This allows conditions that cannot be covered using the structure of the CPN to be catered for. The curriculum definition and representations are independent of the content used to teach them. In this approach the CPN is used to drive the direction and measure progress of the learning, it is the content which delivers the learning to the individual.

### *B. Content*

Just as a teacher can teach the same concept in many different ways, this framework allows multiple pieces and types of content and resources to be available for each knowledge item in the curriculum. The design is content agnostic. It is applicable in any learning domain and can deliver learning content in any format. Currently this framework has been implemented in the \Realizeit platform in fields as wide ranging as Mathematics and Engineering, to Criminal Justice and Psychology, and with content ranging from text and audio to video and interactive animations.

### *C. The Artificial Intelligence Engine*

The AIE is responsible for discovering and adapting to each individual learner's changing abilities, behavior and preferences. It must manage its own accuracy and performance to ensure the individual receives a complete personalized learning experience. It is responsible for emulating what a good teacher would do in a one-on-one situation. To achieve this, the engine must be both intelligent and adaptive. The intelligent side must learn about the learner, content, curriculum and the system itself, whereas the adaptive side takes into account this knowledge to behave appropriately for each learner or group of learners, [6].

## III. Content Framework

The separation of curriculum and content allows each individual to follow a personalized and appropriate learning path independent of the content available to the learner. At each point in a student's progression through a curriculum of study a decision must be made on which is the best knowledge item for the student to study next. Once this decision has been made the AIE must then select the most appropriate content to allow the learner to understand and master the concept.

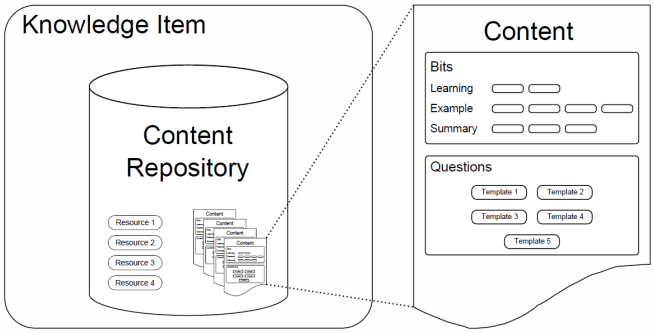


Fig 3. The structure of content in Realizeit adaptive learning platform

In [7], VanLehn proposed that learning and tutoring systems can be considered to be built upon a two loop structure; an outer loop which decides which task the student should do next, and an inner loop which is concerned about steps within a task. The path through a curriculum, which involves at each point deciding on the next curricular item to learn, is part of this outer loop. The selection, adaptation and guidance of the learner through the content can also be considered to be part of the outer loop, and it's the structure of the content which allows this to function effectively and efficiently. This will be discussed in the remainder of this paper. The inner loop which aids the learner as they progress through tasks and questions within the content will not be discussed here.

In order to appropriately adapt learning content to an individual, a generic flexible content structure is required. As this framework is designed to be content agnostic this structure must be capable of dealing with content of any kind and from any field.

Against each individual knowledge in a curriculum item is a content repository. This holds a list containing one or more pieces of content (files) which are available to teach the current knowledge item. A single piece of content may be in several content repositories allowing a many-to-many relationship to exist between content files and curricular knowledge items. Each single content file is broken into learning material and questions. In addition to the content files, a collection of additional resources, such as URLs, PDFs, presentations, audio files and video files can be added to the content repository and made available to the learner. This structure is summarized in Fig 3.

The learning material within a file is divided into parts known as Bits. These are the smallest pieces of learning content that a learner can use, and can include learning sections, examples, worked examples, summaries, etc. An author can provide multiple Bits of a single type within the learning material. Questions are held in a question store. This is not a bank of static questions but rather a collection of generic question templates. Fig 4 outlines the decisions made when it selects, renders and delivers each component of the content to an individual learner. In the following subsections each of

these components and decisions will be discussed in more detail.

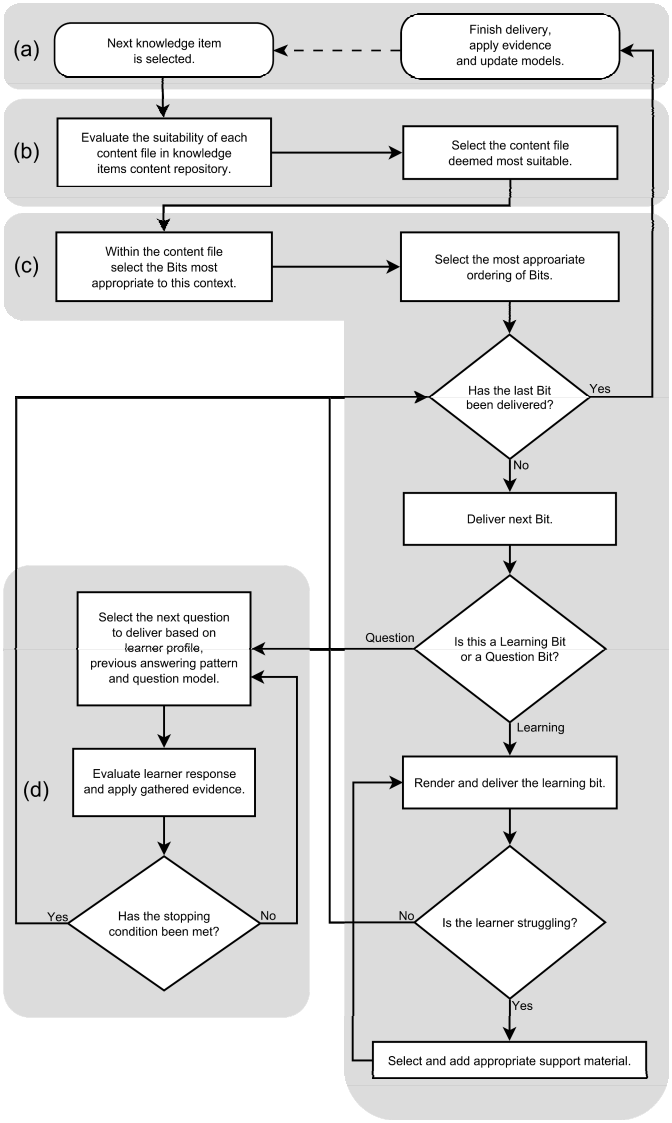


Fig 4. An overview of the content decision process: (a) Knowledge item selection, (b) Content selection, (c) Learning Bits, (d) Questions.

## *A. Content files*

In the \Realizeit implementation of this framework the content files are packaged and managed using a construct that allows dynamic content to be defined and generated in real time for students. It contains tagging to link to knowledge items, and advanced features that would normally require programming e.g. unfolding of information, pop-up references etc.

Associated with each content file is a content profile. The profile contains a model which identifies the situations and subsets of learners for which this content file is appropriate. The content profile can be created by the author during the authoring process. However the profile may not always be supplied, it could be inaccurate or the content may be suitable in situations unforeseen by the author. Therefore the AIE must be capable of deriving and evolving a content profile as data is gathered on the file's usage.

### *1) Author supplied profile*

The author can supply a list of learner attributes that explicitly specify the variables on which the suitability of this content file can be judged. Additionally, they can supply variable expressions which allow the author to create complex expressions that contain many attributes with conditions on each of those attributes. For example an author might supply the attributes *gender* and *learning style* and/or could supply the expression that specifies the content is only suitable for *male tactile learners*.

### *2) Data derived profile*

The author is not required to supply this information with their content. In this case we can assume that this piece of content is suitable for all types of students. However, after using this content a number of times it may become apparent that it works better for some learners than others. The AIE must recognize this and derive and adjust the profile so that it describes the patterns in the usage data.

## *B. Content Selection*

As discussed, for each knowledge item in a curriculum the structure can have several pieces of content available for a learner to use. The AIE must evaluate each content file in a repository and decide on the most appropriate file for an individual learner, at all times trying to maximize the probability of successful learning. This involves matching the learner profile to each content profile. This is not a straightforward process as each profile may hold information not available in the other. The adaptive algorithms of the AIE can handle this missing data. Once a content file has been chosen the system must then render and deliver this file to the learner, Fig 4(b).

During the content rendering, directives which affect flow and interaction can be implemented using a purpose built language which has access to both the learner and content profiles. The rendering engine instantiates the content for the learner and allows for continuous adaptation and learner support during content delivery.

## *C. Learning Bits*

Within its structure, a content file is broken into learning material and questions. The learning material is then further divided into *Bits*. These are the small pieces of learning material that a learner can use, such as an example or summary. At the highest level there are two types of Bits; those that contain questions, *Question Bits*, and those that don't, *Learning Bits*. Each bit is also assigned one of eight different roles: Introduction, Interactive Example, Learning, Questions, Example, Summary, Worked Example and Review.

The processing and rendering of Questions and Learning Bits follows the same logic until the point where the questions must be delivered. This then spawns off a secondary process. This will be discussed in the next subsection, see Fig 4(c)(d).

A single content file can have several Bits of a single type or indeed no bit of a particular type. This granularization of content allows the flexibility to vary a single content file to fit a

wide variety of learning contexts, situations and individuals. For example, different sets of Bits can be used in different circumstances; the order of the Bits can be altered; alternative versions of the same Bit can be included; Bits from prerequisite items can be included if needed, see Fig 5.

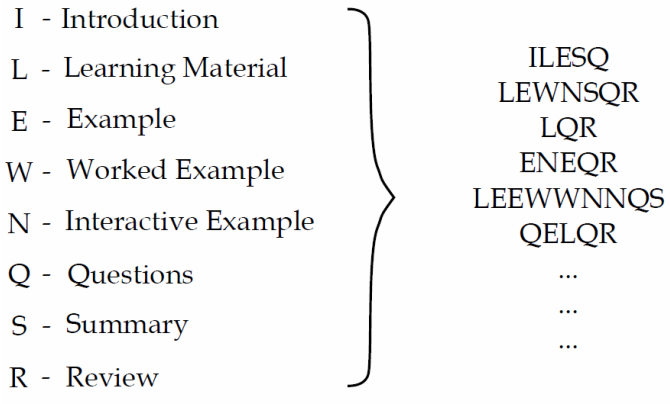


Fig 5. Learning Bits

Not only can this be carried out by the AIE, the learner is free to add, remove and reorder the Bits as desired, leaving ultimate control of the learning in their hands. The data mining and machine learning processes of the AIE must monitor the learner outcomes and extract a model of how, when and where to use the Bits so that they are effective and efficient.

*D. Questions*

Traditionally learning systems have provided banks of static questions to present to a learner, with an over-reliance on multi-choice questions. There are several drawbacks to this approach. The creation of large banks of static questions is time consuming and only provides a limited supply of opportunities for a learner to practice. Multi-choice questions are really only suited to lower-level skills. They can be ineffective in gauging partial knowledge and provide a reasonably high chance of guessing the correct answer.

In the \Realizeit implementation wide range of question types such as enter answer, ordering, matching, mathematics input, point and click and attachment are enabled. Composite questions can also be constructed which can include any combination of available types. Questions are defined using generic data structures which are used to represent question forms. The author of a question creates a form or `template' which is used by the AIE to generate any number of instances of the question, rather than specifying static questions, see Fig 6.

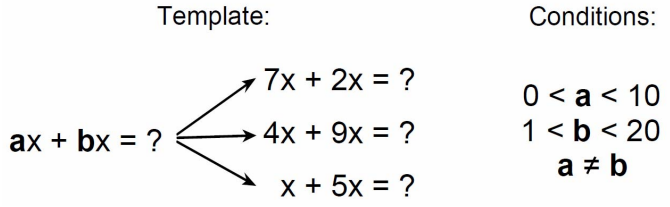


Fig 6. A sample question template

Variable and adaptive questions allow learners to attempt questions which will be generated in real time with different variable values. This allows the learner to have a seemingly infinite number of chances to practice a question, with the assurance of knowing that the same question will not be presented again, Fig 4(d). These variable questions are not tied to a specific content delivery, but can be re-used in different contexts: in a lesson, during revision, when practicing or in an assessment. Additional information and options including hints, question behaviors and structured solutions, all of which aid the inner loop processes, can also be defined.

In addition to the Content Profiles and the model of content Bits mentioned earlier, the AIE builds a set of Question Models for each question in each content file. These provide various metrics which describe the observed patterns in the question usage data such as difficulty, discrimination, probability of guessing and various time parameters. The models allow the AIE to adapt the questions it offers to learners and to provide feedback to the authors so that they can identify specific issues with the questions (e.g. too high a chance of guessing the answer or making a mistake).

*E. Resources*

A content repository may also contain a variety of resources which are available for the learner to use if desired. These range from audio and video files to interactive animations. The AIE tracks the learner's use of these types of resources and infers their preference for them. This can help enrich the learner profile and aid in the selection of content.

## IV. Conclusion

As in any learning system, content is key. This paper outlined a framework which allows the delivery of personalized adaptive learning content. In this structure the content is highly granularized to allow the learning system the flexibility to vary a single content file to fit a wide variety of learning contexts, situations and individuals. Question templates are used to generate any number of instances of a question, rather than specifying banks of static questions.

An Artificial Intelligence Engine manages the interaction between the curriculum, content and learner. It is responsible for building a model of each component of the learning content, evolving and managing the curriculum models and structures, and discovering and adapting to each individual's changing abilities, behavior and preferences.

This framework has been implemented as part of the \Realizeit learning platform. Currently this platform is in place in several large institutions delivering courses, content and gathering data. Future work will utilize this data and concentrate on the refinement and validation of this framework as well as the algorithms and methods of the AIE.

## References

[1] Allen, I. Elaine and Seaman, Jeff, “Grade Change: Tracking Online Education in the United States,” Babson Survey Research Group and the College Board, 2014.

[2] Graf, Sabine, “Adaptivity in learning management systems focusing on learning styles.” PhD thesis - Vienna University of Technology, 2007.

[3] Jonassen, David H. and Grabowski, Barbara L., “Grade Change: Tracking Online Education in the United States,” Handbook of Individual Differences, Learning, and Instruction, 1993.

[4] Shute, V. J. and Zapata-Rivera, D., “Adaptive technologies,” 227-294, Handbook of Individual Differences, Learning, and Instruction, 2008.

[5] Bienkowski, Marie and Feng, Mingyu and Means,Barbara, “Enhancing Teaching and Learning through Educational Data Mining and Learning Analytics,” https://www.ed.gov/edblogs/technology/files/2012/03/edm-la-brief.pdf, 2012.

[6] Vandewaetere, Mieke and Clarebout, Geraldine, “Advanced Technologies for Personalized Learning, Instruction, and Performance,” 425-437, Handbook of Research on Educational Communications and Technology, 2014.

[7] VanLehn, Kurt, “The Behavior of Tutoring Systems,” 227-265, 16(3) Int. J. Artif. Intell. Ed., 2006.