Automating an eLearning System – A Case Study

Sridhar Chimalakonda
International Institute of Information Technology, Hyderabad,
sridhar_ch@research.iiit.ac.in

Kesav V. Nori
TCS Innovation Labs for Business Systems, Hyderabad,
kesav.nori@tcs.com

Abstract
Developing and maintaining eLearning Systems for multilingual requirements manually is difficult in terms of cost, quality and productivity because of the enormous number and huge variety of products. We propose an approach that emphasizes on standardization, tools and processes to enable automation in these kinds of systems. This differs from traditional approaches as it focuses on building systems that involve little or no computer programming. The approach and concepts are illustrated using a large scale industrial project. The lessons learned from this exercise are used to solve “a class of problems” in eLearning Systems.

1. Introduction
India cannot afford illiteracy which is an impediment to development. Tata Consultancy Services (TCS), an Indian Software House has initiated a multilingual eLearning system [1] as a solution to the illiteracy problem under Adult Literacy Programme (ALP) initiative. This solution employs Computer Based Functional Literacy (CBFL) [2] as the underlying technology to develop the eLearning system for a particular language. To teach any of the 22 Indian languages in any other Indian language, we need 22 X 22 products and considering 3 dialects on average for each language, the number of products to be developed amounts to 22 X 3 X 3 X 22 (4356). The scope, size and variety of CBFLs to be developed become enormous as the number of products to be developed increases based on the feedback from CBFLs deployed for use. But the development and maintenance of a single CBFL in itself is a daunting task considering the complexity and ever-changing requirements. CBFL is based on sound principles from the ‘theory of cognition’, ‘laws of perception’ and pedagogic findings and is highly successful as an eLearning system but it’s the development and maintenance of CBFL at the mass-scale is the core problem. Given the social impact of these products, the need to develop them at a less cost, high quality and in less amount of time is inevitable.

The National Literacy Mission (NLM) [3] had already researched the problem of adult literacy for India, devised its uniform approach and produced primers (core teaching material) for each of the Indian languages that embodied the approach. The scope of CBFL is to convert this wealth of instructional material in printed form into multimedia form and our approach is concerned with automating the construction of CBFLs at a mass-scale. Automation is achieved through standardization of product structure, production process and raw material. New products are built from existing raw material and the standardized infrastructure by enabling standard processes. This approach is detailed in section 3.

The current approach and the challenges posed by the current approach are outlined in section 2. The results of our approach are briefed in section 4. Section 5 provides a discussion on related work. The lessons learned through the exercise are summarized in section 6. Section 7 concludes the paper with a perspective on how to solve “a class of problems” using automation in eLearning systems and its application in other multilingual societies.
2. Manual approach to develop CBFL eLearning system

TCS has developed software to address adult literacy concerns in India. As part of this, the development of CBFLs for 8 languages is done by 8 teams at 8 different TCS locations in India. We have found that the development of multimedia material for CBFL is sporadic and unstructured. This spans across development of individual elements, eLearning content and technology usage among others. This led to the following issues & challenges.

- The products that are developed by various teams do not have consistent structure.
- The product development process varies from one product to another product.
- The cost of the product and the resources required to develop the product are high and vary from one product to another product.

Using the current approach, for every new CBFL to be developed, a new team had to be formed, the ideas explained and technology development monitored. It was evident that developing CBFL itself was becoming a bottleneck in creating and maintaining the technology. This approach slowed down the progress of ALP initiative forcing us to think of an alternative. We attack the problem of developing CBFLs as “a class of problems”.

3. eAutomation: Automated approach to multilingual eLearning systems

The challenges posed by the manual approach while developing multilingual eLearning Systems are handled in our approach by enabling automation throughout the development process. During our experience with the case study, the concepts of standardization, process automation have evolved. Each of these concepts along with their application in our case study is explained in the following section.

Figure 1 shows the development process of CBFL using our approach and this process is standardized for developing all products under the ALP initiative. Each of the process steps emerged as we were trying to solve a specific challenge while developing CBFLs. The process starts by understanding and analyzing NLM primers. The navigation model is modeled as a finite state machine that describes the flow between various entities (chapters, lessons, and scenes). This is the solid foundation for production process standardization.
and automation. Part of the navigation model for a typical chapter is shown in figure 2 and [4] explains it in more detail. Standardization and tools provide a solid background to create new products using assembly processes. These processes are modeled using eScript [5], an in-house developed tool at TCS Innovation Labs for Business Systems, Hyderabad and other script technologies.

The next step involves standardization of product structure and raw material as shown in figure 3. This solves the challenge of creating and maintaining a consistent structure across products. Production process is standardized based on the navigation model. The components that are common to all products are identified and developed. The component repositories enable reuse of various elements across products. The next step involves the creation of product structure that consists of a huge number of scenes. Automation tools are developed that support the assembly process. In our case study, a tool was developed that creates the massive product structure (240 scenes and 12,000 elements) and organizes the scenes into chapters and books. The content that is common in each of the scenes is identified and is created using another tool. The assembly process enables the production of the product.

4. Outcomes

We have aimed at automating the development of CBFLs at a mass-scale. Chart1 shows the details of the results. The x-axis shows the major functionalities that have been automated and y-axis shows the time taken in man-years using manual and automation approach. It takes 0.1 man-years to create product structure (240 scenes - 12,000 elements) for 4356 products using our approach and 12.1 man-years using manual approach. Generating CBFL for different dialects given the base CFBL can be completed in 0.9 man-years for 3256 products as compared to the massive 81.6 man-years. We have generated the product structure for 5 Indian languages and created 2 dialects from a base language. A detailed explanation is given in [4].

5. Related Work

The idea of mass production and industrialization has its roots in the famous 1968 NATO conference [6]. Brad Cox’s work [7] explains software crisis and software
industrial revolution as a solution to this. The case study detailed in this paper has similarities with the classical M X N problem. There are M programming languages and N machines; we need M X N compilers so that all languages are available on all machines. Similarly, we have to develop CBFL X CBFL products. Making each compiler or CBFL is a manually intensive task. The PQCC experiment [8] attempted the automated construction of high quality, highly optimizing compilers but with little success. It showed that “Tools do not make a process!”. Our approach has enabled automation by a production process supported by tools and raw material at every process step. Standardization of various aspects using navigation models and finite state machines provided the solid foundation for the entire approach.

6. Lessons Learned

We have learnt how to automate repetitive tasks in software applications by developing the solution to a “class of problems” as opposed to a single problem. In the case study presented, the solution aims at developing CBFLs for all Indian languages rather than for a single language. We have also learnt that standardization is the key enabler for process automation. This should be supported by tools and standardized raw material at every process step. The automation approach has forced various kinds of standardization. The nomenclature, production process and all the elements of the product are standardized in the case study presented. Standardization is also the core basis for enabling interchange of parts in the assembly process which eases maintenance and upgradation of products. This approach can be applied to develop different kinds of learning material like school education, post graduate education in multilingual societies.

7. Conclusion

The core problem in developing multilingual eLearning systems is to cater to the development and maintenance of huge amount of variants and products. This paper illustrates an automation based approach to solve these kinds of problems. This approach involves huge standardization efforts in product structure, production process and raw material as a basis to assembling products. The paper emphasizes the role of tools in the assembly process and employs an innovative assembly tool for production of products. The entire approach is substantiated through an industrial case study that has huge social impact. The paper concludes that the automation approach proposed can be incorporated to solve “a class of problems” and this thinking should be incorporated in software engineering curriculum.

8. References