



JISC Final Report

Before completing this template please note:

- *The Project Management Guidelines explain the purpose of final reports.*
- *Fill in the information for the header, e.g. project acronym, version, and date.*
- *Prepare a cover sheet using the cover sheet template and attach to final report.*
- *This template is for completion by JISC funded project managers*
- *Text in italics is explanatory and should be deleted in completed documents.*

Title Page

*Poetiks final report.
Greg Garrard
4th November 2010*

Table of Contents

<List the topics covered and page numbers.>

Acknowledgements

Poetiks was funded by the JISC Learning and Teaching Innovation Grant, with project support from David Kernoghan. Seedcorn funding was provided by Bath Spa University. Valuable feedback was received from Robert Watt (Dundee) and Stuart Lee (Oxford).

Executive Summary

The JISC-funded phase of development of Poetiks was an ambitious attempt to develop a web-based application that would help teach analysis of poetic technique, especially scansion (i.e. detection and analysis of poetic rhythm). In particular, it was intended to make visible and salient the idea of a poem as a composition of numerous *layers* of meaning, each of which could be considered separately, but which would nevertheless interact with each other under intensive scrutiny. A key aim was to ensure the application would be easy to use, attractive and accessible to users, most particularly students of poetry aged 16-23.

The interface was developed using Flash, so as to allow for the inclusion of attractive visual elements. Moreover, Flash Player is a widely-installed component of web-browsers, and is free and easy to update. We opted for a web-based application at this stage because we were aware that users on open access computers in universities and schools would not be allowed to download an executable application. Flash is particularly preferred by web developers who want a clean look and a simple interface, and who enjoy the relative ease with which the interactive space can be redesigned. The database in which marked-up poems are stored has been developed using XML. This has made searching fast, and also ensures simple compatibility with the MySQL database aspect of the software and any potential future uses.

Alongside the development of the interface, we worked on the core element of Poetiks' functionality: the analysis of syllable and fixed stress (i.e. the regular stress patterns found in English multisyllable words). [Explain what happened here?]

We did not expend effort on the design of the interface, nor on its role and function within the larger website envisaged for Poetiks. While we have always envisaged Poetiks as a smart, attractive and intuitive application, integrated into a more expansive pedagogical site, we considered that we needed

Project Acronym:
Version:
Contact:
Date:

to work from the core functionality outwards. Considerable effort therefore went into such matters as the XML database and the syllabification algorithms.

The core functionality is not, at time of writing, fully stable or operational. As a result, alpha testing has tended to focus on the bugs users have encountered, alongside the limitations of the interface and the learning environment. As the funding period concludes, we have therefore decided to prioritise the latter, so as to perfect the website and interface, then address the remaining problems of functionality and stability. We will then be in a position to beta test, and to develop the supporting screencasts and sample poems that will help users make the most of Poetiks.

The JISC funding may have run out, but development of Poetiks is not over. We have gained additional innovation funding from SWRDA through Bath Spa University to enable us to explore a commercialised version (Poetiks Pro) suited to use in large educational institutions, and we are working with a design company on a brand identity and interface design that will be more attractive and effective. The branding exercise is helping us deal with the fundamental issues that our project has opened up. The Innovation funding is our first step in the sustainability of the software. Creating a commercial version will allow us to keep a freely available version of Poetiks available to individual students around the world.

We aim to release Poetiks to the public in early 2011.

Background

Poetry is a core element of English Literature from primary school to degree level. The central pedagogical problem is to communicate the character and significance of its *difference* as a genre from fictional prose (most students' favourite) without 'technique' becoming a repetitive and routine basis for analysis. Although many English Literature students are passionate and knowledgeable about poetry, the majority evince fear of poetry, seeing it as a fiendish code devised to disguise the real meaning of the poem. The teacher is often regarded as a proxy for the author, cruelly withholding the truth from students. Alternatively, poetic meaning is seen – equally wrongly – as wholly subjective.

Those who go on to become teachers may teach poetry reluctantly and, in some cases, poorly, creating a cycle of anxiety, demotivation and low achievement. A recent OFSTED report, 'Poetry in Schools', assessed poetry teaching as at least 'satisfactory' across all schools surveyed, but pointed out that it tended to be worse than other areas of English provision, and said that 'many teachers, especially in the primary schools, did not know enough about poetry'. Only eight out of eighty-six had 'outstanding' poetry programmes. The worst examples of teaching used 'didactic' approaches orientated towards tests and exams, while the best introduced a wide range of poems and adopted 'active' teaching approaches. A blend of creative writing and analysis was recommended for maximum benefit.

The situation in HEIs is also troubling:

- ⇒ Professor Overton (Loughborough) published the results of a survey of the teaching of versification in 'English' vol.57, no.2019 (2008), pp.266-82. He found there was 'a widely shared belief among higher education teachers that knowledge of prosody is important, and an equally widely shared perception that most students entrée higher education without it', and concluded that 'many remain functionally illiterate as readers of verse' at graduation. It is crucial both to teach the technical terminology of versification in an unthreatening fashion, and to enable students to connect the details of poetic technique with questions of meaning.
- ⇒ Professor Regan (Head of English, Durham) observed the disparity between the increasing public profile of poetry and its attenuation within a literary education, concluding that 'we have to rethink the way that poetry is currently being taught'. (ESC Newsletter 2, August 2001)
- ⇒ Dr D'Agostino (Head of English, Queen Mary's UC) argued that a poem should be 'listened to and appreciated for what it is in its own linguistic terms' rather than immediately being referred to a larger historical or theoretical issue. (ESC Newsletter 5, April 2003)
- ⇒ The English Benchmark says that students should have 'knowledge and understanding of the distinctive character of ... poetry', but as Dr Nicole King (Academic Coordinator, ESC) has

Project Acronym:
Version:
Contact:
Date:

pointed out on the ESC website, students are often hostile to poetry, seeing it as “difficult” or alien, and therefore uninteresting’.

Poetiks is intended to be an exciting and sophisticated application for secondary and tertiary level, using active learning principles to teach poetic form and technique, and potentially to enrich creative writing.

The pedagogical approach adopted for ‘Poetiks’ is based on the project manager’s ten years of experience teaching a level 2 poetry course. Paper or OHP slides are inadequate to represent the complexity and density of techniques and features found in a quality poem, and of course are incapable of generating automatic feedback. Test functions built in to VLEs can provide crude feedback, but cannot be customised by learners, and provide no annotation tools. Interactive whiteboards allow for a degree of annotation, but are not accessible to learners outside the teaching room.

Aims and Objectives

The aim is to develop an attractive, scalable, web-based application to support the learning, teaching and assessment of poetry. Students and teachers will be able to cut and paste any poem in modern English into the program, and develop a complex analysis through a series of ‘layers’. While only one layer will be visible at any one time, the user will be able to switch easily between layers using a system of tabs. If possible, each layer will adopt a visually-distinctive annotation system (e.g. varying fonts, font colour, highlighting, typographical marks, etc.); it will then be possible to view the poem with all annotations (but without text boxes, which would be too confusing) at once, allowing the user to see the poem as a ‘system of systems’. The idea is to make visible the way that different kinds of *patterns* interact in the reading and analysis of a poem.

At the same time, each layer will incorporate pedagogical elements, and the application will include a number of complete worked examples (of poetry out of copyright). For example, because students often find it difficult to recapture their own instinctive grasp of rhythm once they start to analyse it, the Rhythm layer will include a facility for users to record their own reading of the poem, and play it back as they scan the text. The Words layer will allow teachers selectively to blank particular words, encouraging students to focus on diction by making their own choices (Cloze procedure).

Poetiks will:

- ⇒ Highlight the generic distinctiveness of poetic as a verbal art.
- ⇒ Support the learning, teaching and assessment of specific poetic techniques such as metre, phonemic patterning and lineation.
- ⇒ Allow customisation by users, who will be able to add, remove and alter layers.
- ⇒ Promote rich annotation of the poem by individuals or groups for use in lectures, groupwork and assessment.

Most of these aims have been partially or wholly achieved. We decided not to include a Words layer at this stage, because the ideal (interaction with the OED) proved impracticable, and so the idea of including Cloze procedure was shelved for possible inclusion in Poetiks Pro. The interface redesign and development of the wider website will help to clarify the pedagogical intention of Poetiks, which at the moment tends to be lost on users. A crucial question we have yet to resolve is how to make the techniques identified in each of the layers *visible* to the user in an intuitive way without cluttering up the screen too much.

Methodology

We had originally intended to follow a formal project management programme, but due to the small size of our team (3 members) and the regularity of our meetings (at least every fortnight) it seemed unnecessary. We were constantly discussing progress and reviewing ideas, although there was a period in the middle of the development process when the Project Manager, who has no programming expertise, was somewhat remote from the project. During that period, Anthony Head and Andy Bevan worked closely together to develop the interface and the core functionality.

Project Acronym:
Version:
Contact:
Date:

When we began user testing, we set up Survey Monkey questionnaires to gather responses from the expert user group and from remote student users. In practice, though, bugs with the programme limited the amount of user feedback we could obtain.

Implementation

The first part of the project involved investigation of techniques and the development of a software specification. We knew that we could keep thinking of ideas of features for Poetiks, but decided to limit the content to particular features, namely: syllabification, fixed stress detection, layers for rhetoric, phonemes and imagery, essay layer, loading/saving of poetic analyses, pasting in of poems, note annotations.

By the new year (2 months in), we had created our specification and settled upon the features that we were trying to implement. We then began to create the background functions whilst investigating the methods for the most difficult aspect of the project, accurate syllabification and fixed stress detection.

We were pointed to a method for syllabification described by Paul Holzer, in a 1986 edition of Byte Magazine. This method was adapted in previous research by Charles Hartman, for his software Scandroid. We purchased an original copy of the magazine and were able to interpret it in Actionscript. We had set ourselves an aim of minimizing the use of a dictionary, in order for us to fully develop the syllabification methods.

The syllabification process has been difficult to get right, as expected. In addition to the method described above, which was complicated to implement, it left many words that didn't syllabify correctly. Thus, we weren't getting anywhere near the accuracy that we had hoped for. We persevered with the method, by adding a substantial list of exceptions to the rules by searching for patterns of letters that don't comply with the method. This has led to a fairly accurate method of syllabification, and at the time of writing we are finessing it to get it to an acceptable level of accuracy for public use and trust in the software. A small dictionary is being used to cater for words that don't syllabify correctly.

The other automated process was detecting fixed stress. The process we've adopted for this is to study words in order to examine where and when certain types of words are stressed. Fixed stress only occurs in words with more than one syllable. This method is not accurate enough yet, although we are still improving it. We had been trying to do this solely on character patterns, and we realised the need to add the detection of parts of speech to enable further accuracy. We may never get this method 100% accurate, but as we are creating a learning tool, the highlighting of words that may have multiple ways of being stressed, may prove to be a better pedagogical practice.

By the summer we had implemented most of the functions and a method for overlaying the layers that form the analysis.

Formal testing began much later than had been intended. The timing of our project required alpha testing to begin well before the summer when students were still around, but we had not made enough progress by then to make it possible. So we continued to improve the software over the summer and began formal testing in October, with our expert group and our class of students.

Testing has emphasised the need for accuracy in the automated functions, as well as providing many points for us to consider about the overall interface.

We have engaged a design agency, to help deal with some of these issues of design in order to get the software interface correct. We realised that a full branding was needed in order to fully understand the product that we have created, its audience and its potential.

Outputs and Results

We have expanded the method of syllabification described by Holzer, adapted by Hartmann. By adding a list of exceptions we have created a method that is reasonably accurate without the use of a dictionary. We are in the process of publishing our variation of this method in order to gain more

Project Acronym:
Version:
Contact:
Date:

debate and enhancements. We have not yet formally tested the accuracy of this method, but it is approximately 99% correct, in terms of spoken syllable detection.

The stress detection methods we are using are still in an experimental stage. When compared to non-dictionary methods of detecting stress however, they are appearing quite favourable (with a greater than 90% accuracy). When fully established and improve we will publish these methods for further scrutiny and enhancement.

We presented a paper on the project at the Education and Information Systems, Technologies and Applications (EISTA 2010) conference. This was a good opportunity to discuss the project with colleagues from around the world.

We have a website that contains the project blog (www.poetiks.net), which documents the ideas and decisions we discussed on the way through the project. This website will be converted into the hosting website for the software, and become a social portal for debate around poetry.

The main output of the project is the software itself, currently, still called *Poetiks* (but the branding exercise we're going through may change this). The alpha version of the application can be found on <http://www.poetiks.net/testing/poetiks.html>. It shows how the layers function (when a poem is input by pasting or loading). Our testing has led to the realisation that the layers aspect is not actually obvious to the new user, as the layers are empty in a non-marked up poem. The layering becomes apparent when a poem is marked-up with highlights and notes. This issue is something that we will address in our updated version, before full public release.

Outcomes

We are disappointed that *Poetiks* has not reached a point where it is stable enough to release, or attractive enough to potential users for its value to be appreciated. Several improvements will need to be made for the impacts to be realised: the brand and site will need to be thoroughly revised; the interface will need to be made much more attractive and intuitive; and the core functionality will need to be fully stable and reasonably accurate. While we would have hoped that the last two of these would have been achieved by now, we have successfully bid for funds to carry on with the project, the team is still enthusiastic and working together, and *Poetiks* still inspires us with its possibilities. Assuming all three elements come together, we envisage *Poetiks* modelling a new practice of technology-enhanced '21st century reading', as well as forming the centrepiece of a pedagogically-orientated site for teachers of poetry.

Conclusions

We always knew that we were working on a tight budget in terms of time for the project dictated by the limit of the funding. The project has not kept to its original schedule, as getting the software to a publicly releasable state has taken longer than expected. This is due in part to the complexities of the syllabification and stress detection process, but also mistiming the sequence of work packages. However, the mistiming is in part due to the need for accuracy in the two main automatic processes.

We realised that bugs and inaccuracies actually put off our willing testing audience to a certain extent. The students were not paid software testers, and hence their incentive to persevere with the software was not great. Some of our expert testers were decidedly underwhelmed by the software that we showed them, but our web statistics shows that they may not have spent all that long testing. However, other experts were more positive, and a small number of students on the Poetry course that gave rise to *Poetiks* use it in preference to paper-based analysis. Having overcome some of the problems that confront the first-time user, they report that *Poetiks* is significantly quicker and more accurate than scansion on paper. Nevertheless, improving the accuracy and including a social networking aspect to the software will considerably help the projects cause and answer some of the helpful criticisms provided to us.

The concept of what we are trying to do has regularly been met with considerable enthusiasm, so we just need to make the appropriate adaptations to make the project a fully sustainable success.

Project Acronym:
Version:
Contact:
Date:

Implications

The project should have profound implications for the teaching of poetry analysis. We are calling this '21st century reading', mixing technology with traditional classroom methods. The Poetiks website will also become a unique resource to support the *teaching* of poetic technique, a topic that has received surprisingly little attention. Once we have grappled with the complex questions of copyright and plagiarism involved, we hope to make Poetiks a social learning environment, in which students and teachers can easily compare responses and analyses. Poetiks Pro will introduce additional features and layers, including (potentially) automated assessment. The methods used could also be adapted for students learning English as foreign language, or even help children to read.

If our results are substantiated then the accuracy levels for our automatic processes may well be better than that achieved before, which will have an impact on the Computational Linguistics community e.g. in the development of text-to-speech systems.