**MARECKI — Renderings**

<0 images>

Renderings: Translating Literary Works in the Digital Age

Marecki, P. and Montfort, N.

The point of departure for this paper is the Renderings project (http://trope-tank.mit.edu/renderings/) established in 2014 and developed at the Massachusetts Institute of Technology in a lab called the Trope Tank. The project is described as concentrating on translations of highly computational and otherwise unusual digital literature into English. Its members ‘not only employ established literary translation techniques, but also consider how computation and language interact. Literary and computational experts worldwide participate’. The current team includes Nick Montfort (the initiator and leader of the project), Patsy Baudoin, Andrew Campana, Sally Chen, Aleksandra Małecka, Piotr Marecki, and Erik Stayton. During the project’s first year, 13 translations or bilingual works, by 12 authors, have been produced in the following languages: Chinese (1), French (3), German (1), Japanese (4), Polish (2), and Spanish (2). The translated works are

• Automation (2013) by Andrew Campana.

• Contemporary Japanese Poetry Generator (2012) by Shinonome Nodoka.

• Dizains (1985) by Marcel Bénabou.

• Hallelujah (2012) by ni\_ka.

• MAZ—Mutantist Autonomous Zone (2014) by Mathias Richard.

• Poem 21 (1988) by Amílcar Romero.

• Poet (2003) by Michał Rudolf.

• Sample Automatic Poem (2009) by Féliz Remirez.

• Seika no Kôshô (2013) by Andrew Campana.

• Shanshui by Sally Chen.

• Speeches (1993) by Marek Pampuch.

• Tötan das Gedich (1997) by Johannes Auer.

• Triolets by Paul Braffort.

The programming languages of the original works include Basic, Perl, and Java Script. They were selected to represent the wide variety of genres of electronic literature and creative computing, and the productions of cultures/literatures not currently well known in this dominantly English-language field. Thus, the first Renderings set of works includes genres characteristic of specific cultures, such as Japanese ‘monitor poetry’ (a blog that bursts of flowers, hearts, and other graphics dense enough to obscure the screen), a Polish generator of communist speeches, electronic ‘landscape poetry’ from China, and electronic OULIPO texts (France). The selected works also present different approaches to computation in literature. The project itself thus has the aim of describing the experiences of the margins of digital culture and exploring the hitherto overlooked fringes of the digital heritage.

The Renderings is not the first project exploring translations of electronic literature. There have already been translations of Michael Joyce’s, Stuart Moulthrop’s, Nick Montfort’s, and Stephanie Strickland’s works from English into other languages. In addition, the Electronic Literature Organization was a co-sponsor of the conference Translating E-Literature in 2012 in Paris. The Renderings project continues these threads, but focusing on the direction from other languages into English, its goal being to give English-speakers access to works from other traditions. The project involves also meetings and brainstorming with literary translators: Robert Pinsky, Marc Lowenthal, John Cayley, and David Ferry.

Translating digital works written in code requires the translator to face new challenges in addition to those tackled by the regular translator of literature. It is a type of translation akin to the translation of experimental, conceptual or constrained works. It is not rare that the task requires the translator or translators to reinvent the work in a new linguistic and cultural context, and sometimes also another programming language. The history of literature is already familiar with similar cases, like the translations of works of the French OULIPO group; for instance, Georges Perec’s La dispariton, which is written without the most frequently occurring vowel of the French language, has been be rendered in other languages with the omission of the most frequent vowel in the language of the translator, e in English but a in Spanish.

In the case of highly computational digital works there are additional difficulties and challenges, first and foremost, the formal and material properties of the code of the program. If we assume after Noah Wardrip-Fruin that a digital work has three layers: the input, process and output, the task of translation will be operated mostly on the first two layers: the input and process. It will require establishing a lexicon, determining the input data, which may differ given the discrepancies between grammars (inflection, declension, genre) and translating the process, that is the lines of code responsible for producing a given output. Noah Wardrip-Fruin explains to humanities scholars analyzing digital works that they focus mainly on the output level, which he considers a superficial approach. It seems that the work of the translator of a digital work is the ideal activity for performing what this scholar calls ‘expressive processing’. Translating a highly computational work without the knowledge of its inner workings and operation on all the three levels of analysis should not occur. For instance, the novel World Clock, written by Nick Montfort in Python, has 165 lines of code in its original English version. Its Polish translation has an additional 60 lines of code. Indeed it is not unusual for the output of the translated work to be the result of processes different than in the original code. An important category for the translation of digital works is collaborative work, in a team including a translator and a programmer, where translation and programming competences overlap and complete.

The translation of a digital work is not only a matter of language, but also requires awareness of the code and the platform for which it was designed. Especially in the case of older works, the translator has to consider porting the work to a platform more accessible to the contemporary reader. The textual generator Poet was written in Perl in 2003 and archived online as a .pl file. It will be thus available to those readers who know how to run the program in the terminal and are willing to download and execute it. Given the goals of the project, it seemed justified to port the program to Java Script in order to publish it online, to provide easier access to readers on the Web. In such a case the translator has to preserve as many aspects of the program’s functioning as possible in the original programming language. Yet another aspect connected to this problem is the change of platform. Platform consciousness and platform description are described according to the methodology developed by Nick Montfort and Ian Bogost in books from the Platform Studies series published by MIT Press. It is a method that ‘investigates the relationships between the hardware and software design of computing systems and the creative works produced on those systems.’ And so when describing the generator of communist speeches translated and published online as part of the Renderings project, an important aspect of its historical, formal and material analysis will be the consideration of the consequences of the fact that it was first written in a popular programming language for the Amiga.

The presented paper describes a phenomenon belonging to the broadly understood discipline of creative computing and studies the work of the translator as taking place both in code and language, drawing from the methodology developed by the fields of code studies, platform studies and expressive processing.