Developing Web Databases for Aboriginal Language Preservation

Marie-Odile Junker and Radu Luchian
Institute of Cognitive Science, Carleton University, Canada

Abstract

This article discusses the development of integrated multilingual Web databases to help the preservation of the Native American language East Cree. The creation of digital online resources for threatened aboriginal languages presents many technical, educational and ethical challenges. We focus here on the technical challenges in order to discuss both the problems encountered in this particular context, and the solutions we have considered and explored. We illustrate our discussion with examples from an Oral Stories Database we developed in collaboration with Cree education consultants and speakers in 2002–04. We advocate an approach that includes fast-prototyping, open-source development, and design for the database engine that balances speed, availability, features, and resources. We discuss the impact the combination of this technical approach and the participatory action research method is having on language maintenance.

Correspondence:
Marie-Odile Junker,
Institute of Cognitive Science,
Carleton University
Canada
E-mail:
mojunker@connect.carleton.ca

1 Introduction

The goal of this article is to report and discuss our development of integrated multilingual Web databases to help the preservation of the Native American language East Cree. The creation of digital, online resources for threatened aboriginal languages presents many technical, educational, and ethical challenges. We focus here on the technical challenges in order to discuss both the problems encountered in this particular context and the solutions we have considered and explored. We illustrate our discussion with examples from an oral stories database we developed in collaboration with Cree education consultants and speakers in 2002–2004. We advocate an approach that includes fast-prototyping, open-source development, and design for the database engine that balances speed, availability, features, and resources. We discuss the impact that the combination of this technical approach and the participatory action research method is having on language maintenance.

East Cree is a Native American language of the Algonquian family, spoken in Northern Quebec in the James Bay area. It has 13,000 speakers spread over nine different communities and a vast geographical area. There are two dialects, Northern East Cree and Southern East Cree, the latter consisting of two sub-dialects, Inland and Coastal. In 1995, Cree became the language of instruction from kindergarten up to Grade 3 in all Cree schools managed by the Cree School Board, creating a greater need for teaching resources for language and culture courses. The eastcree.org website was created with the intention to explore how information technology can assist the creation and distribution of Cree language resources. A participatory action research framework was adopted (Morris and Muzychka, 2002; Junker, 2002), which meant (1) that we would focus on the research PROCESS rather than on the research RESULTS; (2) that the success of our research would

Literary and Linguistic Computing, Vol. 22, No. 2, 2007.
doi:10.1093/llc/fql049 Advance Access Published on 8 December 2006

depend on the positive impact it had on language and speakers; (3) that we would define our goals and methods in collaboration with our partners.

A partnership was established in 2001 with Cree programs, a department of the Cree School Board that specializes in creating resources for Cree language and culture courses. We work together to ensure the participation and feedback of speakers, curriculum designers, and teachers of the language. In the process, we are also training willing native speakers in relevant areas such as maintaining the online databases and editing/archiving digital sound records.

Because the Cree schools are spread over a vast territory, a first challenge was to overcome the problem of distance communication. Information Technology seemed perfectly suited for this, but few of the existing tools had been explored or adapted when we started. Another major challenge is that Cree uses a syllabics writing system to which computer technology has been up to now rather unfriendly (Jancewicz and Junker, 2002, 2003). With e-mail and chat rooms becoming increasingly popular among native people, but only available in the colonial languages, we felt that these tools were one of the many reasons that native languages and culture are losing ground to western influence. Another goal of our project was to record and try to preserve what was left of the memories of the elders and of the traditional Cree way of thinking and general world view imprinted on the language with its several dialects.

2 The Web Databases

2.1 A response to the challenges of the participatory action research approach

The web databases accessible at http://www.eastcree.org were developed in order to systematically organize language material and knowledge in culturally sensitive ways. We wanted, for example to preserve the oral tradition, a thousand-year old practice in Cree culture, and felt that Information Technology could offer a support previously unknown. By making old language material available again in an oral form to the younger speakers via the internet, a medium that they are attracted to, we felt that there was a greater chance for language vitality and survival. The databases had to be accessible to all concerned, that is, not only to a few educators, but to all Cree people living in the Northern communities, and also to urban, off-reserve natives wanting to reconnect with their ancestral culture and language. With Cree Programs offices in many different communities, hundreds of kilometers apart, the databases had to allow collaboration at a distance. They had to allow easy modifications, be it for updating the content or for maintaining the interface and the functionality. Finally a lower cost compared to previous ways of doing things (print, paper, postage) was also a priority.

2.2 The eastcree.org tools for language preservation

We started with a publication catalogue (pubcat.eastcree.org), at the request of Cree Programs staff and teachers. Cree Programs has published hundreds of books for schools over the years and teachers needed a way to know what was available and get it to their classrooms. We then worked together on an oral stories database (stories.eastcree.org), in order to digitize, archive, and organize old recordings of elders, recordings that were in danger of being damaged by time, and were not accessible. These databases are multilingual. They display in four languages: East Cree Northern, East Cree Southern, French, and English. Two writing systems (syllabics and roman orthographies) are available. Further developments included a Terminology forum and a Cree (syllabics) Chat room (ayimuwin.eastcree.org). In December 2004, the Cree dictionaries were published online (dict.eastcree.org). Prototypes for spelling lessons, read- and sing-along (lessons.eastcree.org) and a Linguistic Atlas (atlas.eastcree.org) were also developed using the principles we discuss in this article. For sake of brevity we illustrate our approach primarily with one database, the oral stories.

2.3 Example: The oral stories database (http://stories.eastcree.org/)

The oral stories database contains recordings from the Cree School Board, academic scholars,

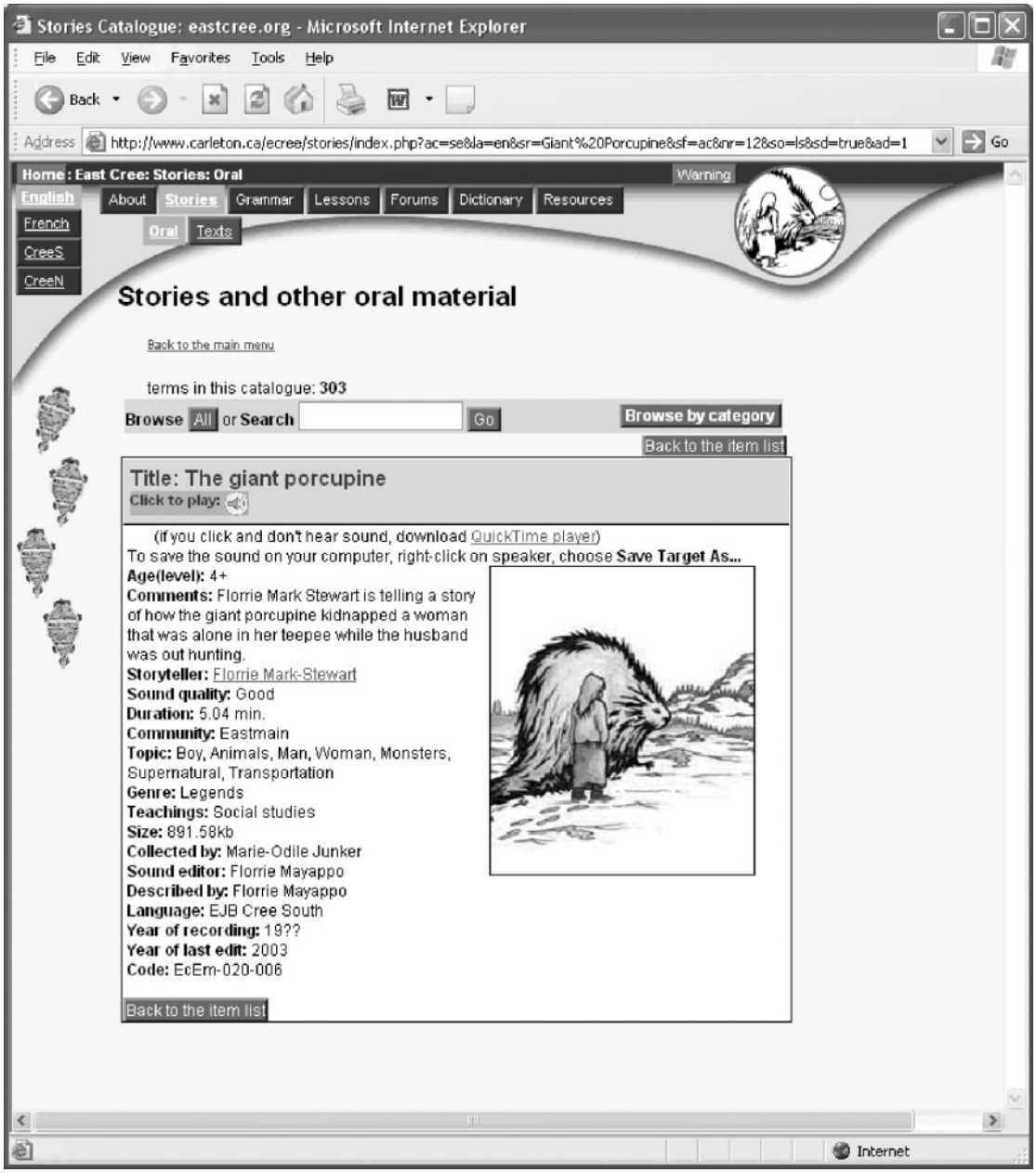

Fig. 1 English item view for the general user

anthropologists and linguists, some community radio stations, and private (speakers) collections. Figures 1 and 2 are screen shots of the database as it is published on the web, and accessible to all users. We call this set of interfaces for general browsing 'end-user interfaces', as opposed to 'maintainers' interfaces' which we will discuss later.

At the Cree School Board, there was no systematic centralized archiving system in place. The tapes were scattered in several communities,

Fig. 2 Cree list view for the general web user

and some had not been backed up. Some were already damaged. The original tapes were between 20 and 40 years old. The same was true for some (speakers' or scholars') private collections.

There was at the same time an increasing need for educational material for the upper grades. Children and teens, having gone through the Cree as a Language of Instruction Program (henceforth CLIP) since 1995, were thirsty for more language material in Cree. The focus of Cree Programs had been, like in many other native communities, almost exclusively on literacy training (Burnaby, 2002) and there was a lack of educational focus on the thousand-year old oral tradition.

The oral database was thus developed as a response to these needs and problems. We wanted a usable tool both for collaboration and for storage of results. And we knew that our interfaces had to work for users with various (mostly low) technical expertise levels. The material had to be organized in culturally relevant categories, and that organization had to be flexible, allowing for online changes without having to reprogram the database or the interfaces. For example, the Cree language distinguishes between stories that are *tipachimuwin* (personal stories, memoirs), and stories that are *aatiyuuhkaanh* (archetypal stories, legends, and myths). Thus our Cree collaborators created in the interface categories that were based on the Cree language, rather than the English language. Because we were working primarily with users from the Cree School Board, the categorization had to reflect the potential use by teachers of the stories, i.e. concerns with age appropriateness, curriculum topics, etc. We also had to take into account the available hardware, speed of connection and software available in the Cree schools and homes.

The design process for the database was developed in consultation with our Cree collaborators. We digitized a first collection of tapes and started to hold a series of Sound Editing and Database Training Workshops. These workshops consisted of teaching our collaborators from Cree Programs as well as various community members who were fluent Cree speakers (radio station technicians, teachers from the Cree schools, and youth from the communities) the basics of sound editing and how to catalogue the oral stories material into the database. Several workshops and some longer work-training sessions for Cree students were held in various locations over a period of 18 months. In total, about thirty persons were trained, with some participating in several workshops. A total of 303 stories were carefully edited and catalogued. At each workshop, new decisions had to be made regarding metadata and interface needs. As the workshop participants were using what we call the 'maintainers' interface', illustrated in Figs 3 and 4, they made suggestions for improvements. Programming was done on site or immediately after, allowing for continuous input.

This item list (Fig. 3) offers a general glimpse into a set of records, and can be sorted by ID, code, date of last edit, size and title. More metadata for each record is available in pop-ups when the mouse cursor is placed over a bit of data (e.g. the editor name pops out of the year of last editing). Since this list is used for content editing purposes, it presents much more detail about each record than the item list seen by the end-user (in Fig. 2).

Fig. 4, the item view as seen by the content editors, corresponds to Fig. 1. The fields which are common across langages (with their lists of values editable as site terminology, Fig. 5) are shown first, followed by the language-specific fields. Thus the record can be seen together, and it is easier for content editors to synchronize the different language versions of the descriptions. This is an innovative approach; existing versions of coding for multilingual audiences present each list of translations separately, with no easy way of comparing between translations.

This separate set of maintainers' interfaces gives control not only to the content of the databases, but also to pieces of layout[1] and even on code elements, as illustrated in Fig. 5.

For example, the 'Edit Terminology' window in the maintainers' interface shown in Fig. 5 below controls the 'Topic list' that appears in the general users' interface shown above in Fig. 2. This is how we were able to build a user interface in Cree even though the programmer on the project does not speak the language. Usually such distributed content-managed systems[2] (dCMS) are being used

Fig. 3 List view of the maintainer's interface

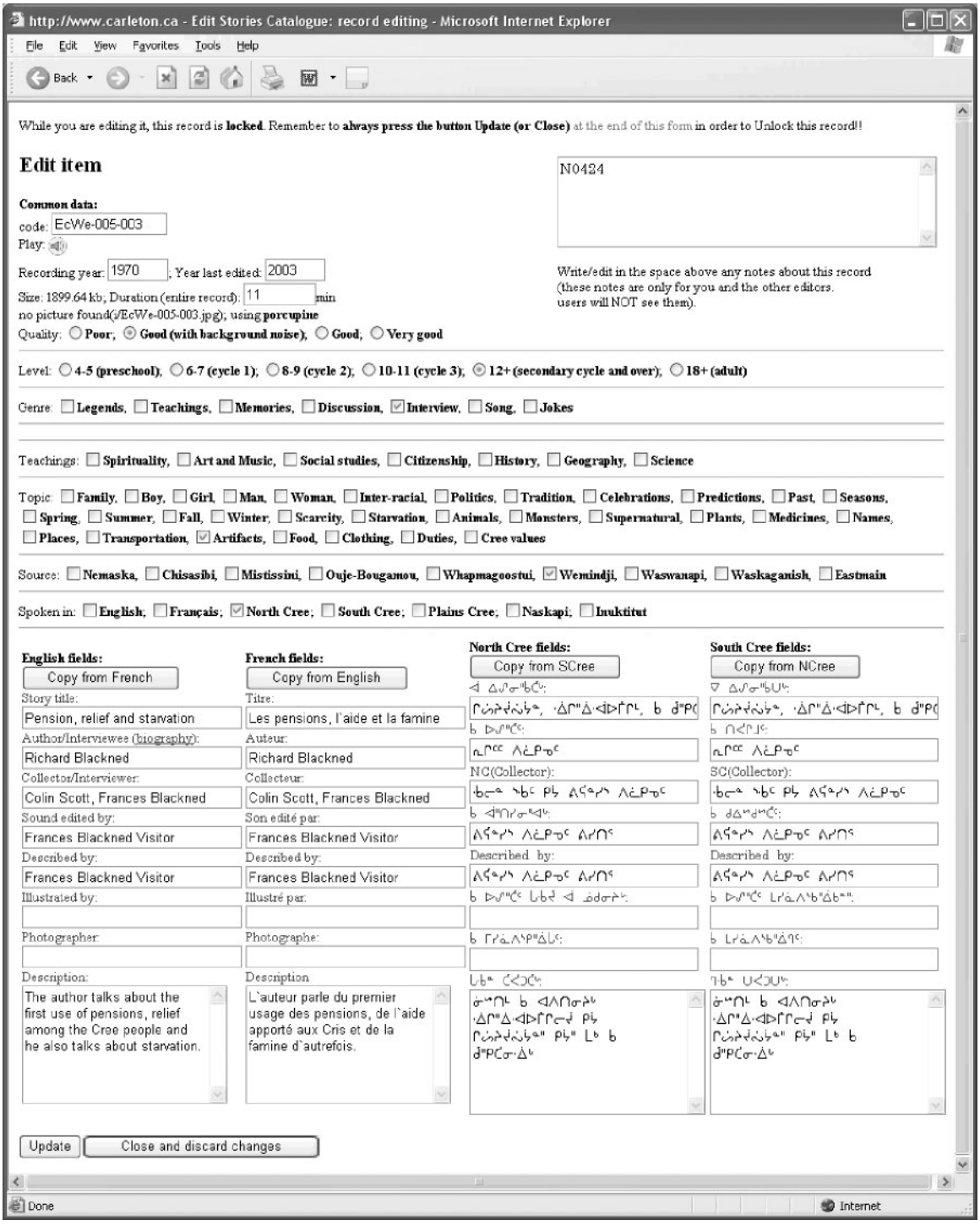

Fig. 4 Edit item view of the maintainer's interface

Fig. 5 Edit terminology view of the maintainer's interface

for technical applications, by programmers or modelers. To apply it to the field of language preservation is an innovation in itself. Even then, what existing dCMS[3] do is manage content and, in fewer cases, layout. Managing code had not been done with this level of user-friendliness even for proprietary systems, when we started in 2002.

The oral stories database is, at time of writing, in a new developmental phase, which attests to its success. Our Cree collaborators have included it in their teacher training interventions, showing teachers how to prepare material for classroom use from the oral stories database. For example, two teachers in each of the nine communities have now been trained to assist others in performing searches and burning CDs for classroom use. Other current classroom applications include research assignments to upper grade students on certain themes (comparing different versions of the same legend, finding out about an ancestor's way of life,

(a)

Fig. 6a Biographies—end-user view in Cree Southern

researching traditional cultural information about a plant or an artifact). A database of the biographies of storytellers is being populated (Fig. 6a, b), and children in school are asked to produce drawings to illustrate each story, drawings which are then being posted on the web.

The biographies are maintained in four languages, available to the content editor in one window.

Finally, new collections from anthropologists are being donated to the Cree School Board to be added to and published in this growing database.

Publishing cultural material on the web can sometimes present ethical challenges such as cultural appropriation issues, copyright infringements, and the violation of privacy. The Cree editors addressed the latter by carefully editing and occasionally censoring some stories or parts of stories. The copyright issues were addressed using the official mandate of our research partner, the Cree School Board, to protect and promote the Cree language for future generations. This mandate is defined in the James Bay and Northern Quebec Agreement, signed in 1975. The Agreement has been interpreted as giving collective rights precedence over individual rights in the case of language and culture preservation. While some anthropologists might encounter difficulties with aboriginal communities or individuals in publishing recordings of their past field work on the web (like it is attempted in the E-meld project), in this case, contributing anthropologists and ethnographers are just asked to give usage rights to the Cree School Board, who is then free to publish the material on the web to pursue its official cultural mandate.

For reasons of space, we cannot describe in similar detail the process behind other web databases on the eastcree.org site. Let us just say that the same participatory approach underlies all developments. For the creation of the reference grammar, the Cree dictionary database, the book catalogue, the terminology forum, or the verb paradigm database (forthcoming), a combination of training and research workshops and ongoing

(b)

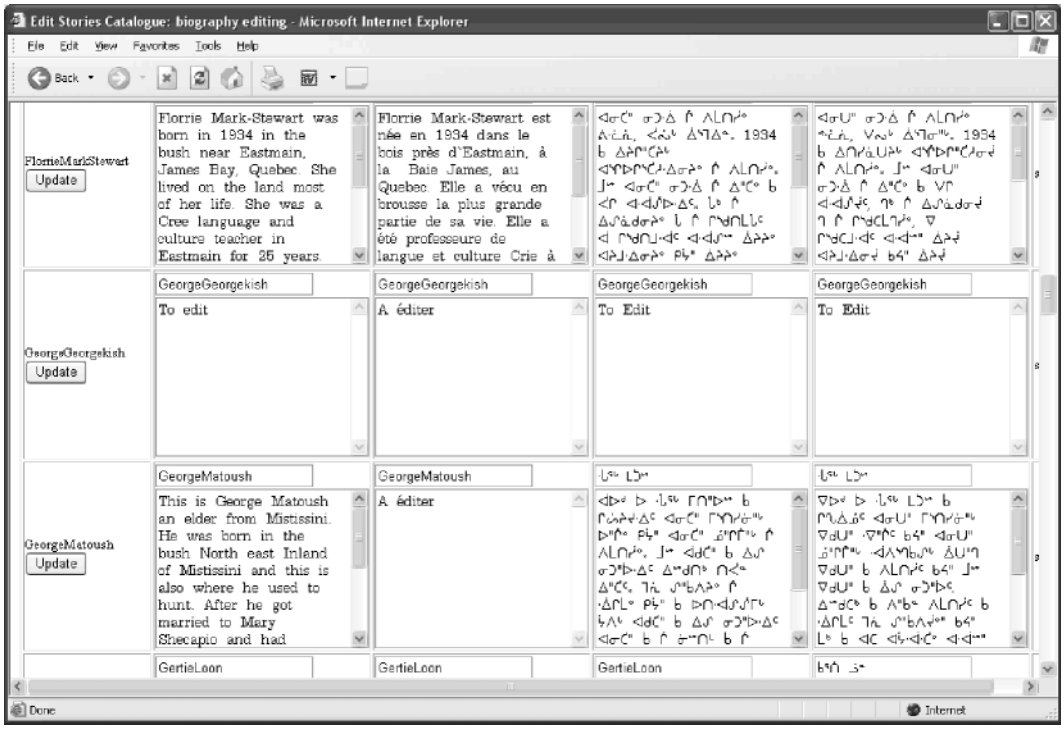

Fig. 6b Biographies—maintainers' interface

consultation during programming have been and continue to be used. Before discussing more fully the general impact of this approach on language preservation, we turn to our design approach and the technical challenges we had to overcome.

3 Design Approach

With the requirements set up by our choice of research methodology (Participatory Action Research, henceforth PAR), and by our choice of presentation medium (Web), we had to address many technical concerns in ways that would not detract from the availability of the data. In this section we review these requirements and the solutions we chose, and recommend some future directions.

3.1 Design principles

We developed the online databases and their interfaces following the principles of a pattern language for document design called MonDoc (Luchianov, 2000).

A pattern language is a structured method of describing good design practices in a particular domain. The concept of pattern language was first proposed within the field of architecture by Alexander *et al.* (1977). It was then adapted to software engineering by Gamma *et al.* (1995), as a methodology for systematically and consistently developing software. In the eastcree.org project, we are developing a collaborative repository for language documentation and preservation that is speaker-oriented. We used the following principles:

- Adaptive presentation (generation of documents from pieces according to user-specified levels of detail).
- Reduced contextual clutter (present the most relevant data, metadata and interface elements).
- Context-preserving data mining (like access to more data in a stack of notes—without having to move to other pages).

- Data access in graph-oriented rather than serial paths (as suggested in Spinellis, 2001).
- Transparent encapsulation of content, layout and code (allowing for a variety of visibility levels of the details of implementation of each element of the application).
- Informative reports (on status and errors).
- Active path markers (buttons with pop-up descriptions rather than text-like hypertext links).

The more useful aspects of the pattern language we are working with are that: (1) most of the components of the databases (code, templates, and multilingual content) are shared; (2) the site is built based on a common set of multilingual terminology content templates, and it uses a common set of JavaScript objects; (3) we continue to modularize, as shared functionality requires it, while keeping to a minimum the number of files to be maintained; (4) multilingual features for:

- Data (of the grammar description, forums, lexica, technical resources, etc.).
- Meta-data (for stories and print publication catalogues).
- Interface terminology, layout, and coding.[4]

One of the most important and innovative[5] distinctions which this pattern language makes is between the *graphic style* and the *semantics* of an element of content. For example a style definition like '%bgcolor=yellow%text%%' gives specific instructions about the look of the text (in this case, make yellow the color of the paper on which 'text' is shown), while a semantic definition like '%highlight%text%%' describes the text for a given interpreter object method, which is then free to make the highlight bold or not, yellow or cyan, on the background or on the text, depending on context. The context can be the target display format (e.g. html, pdf, rdf, general xml), the skin[6] used, another markup[7] enclosing this one, etc. This distinction applies to elements of content one by one or in combinations, as in the 'cascading' element of CSS.[8]

Another important aspect of the pattern language is the inclusion of the development process known in Software Engineering and Human-Computer Interaction literature as *fast prototyping*[9] (Zhang *et al.*, 2003). This alternative to other development styles (especially the ones advocating thorough project planning followed by implementation, as per White, 1994), is better suited to PAR, since features are available for discussion and evaluation before time is invested to finalize them, thus allowing end users and data editors to have a better understanding of the tool and a higher impact on more of its development stages. Our first attempt at fast prototyping was to build an online syllabic communication tool. It evolved into a general transliteration tool for simultaneous entry of syllabic and roman orthography, used for example in a Cree chat room, and in a terminology forum. As far as the databases are concerned, we started to work in the now-classical Object-Oriented paradigm (e.g. White, 1994), spending lots of resources on creating development plans, database structures and object models; we then introduced fast prototyping and worked with the users on a first series of prototypes which resulted in the publication catalogue (pubcat.eastcree.org). This process took more than an academic term (5–6 months), to develop and test. For the second database (the oral records stories.eastcree.org), we used fast prototyping from the start, and due to the reuse-friendly design of the database engine, its prototype was ready in 3 days and finalized in an extra couple of weeks. We reused modules and built the simplest database structure, functionality and interface, and grew from there, as the native users required functionality.

Theories like pattern languages and fast prototyping have allowed us to meet most of the PAR requirements outlined in Section 2. Making such flexible systems available over large distances, in (relatively cheap) collaboration-friendly environments becomes then a more technical matter, which we discuss next.

3.2 Web design challenges

Until recently, computer programs used to be heavily optimized in order to reduce the resources they were using up. However, since the explosion of storage and processing power, size and speed optimization seems to be relegated to computer courses and to the embedded systems where

stringent resource limitations still exist. We live in a time where 'bloatware' (software with a needlessly huge install footprint), is no longer a sin, but a common feature. However, as far as Web applications are concerned, we still have several limitations (Zhang *et al*., 2003): (1) heterogenic client bases (since people of limited technical background tend to disregard upgrading their web browsers—or do not know the available option), (2) low or nonexistent access to most client machines, a problem common to all Web applications, and (3) long transfer times negatively impacting the users' appreciation of the application and its data (since we need to supply data to areas where broadband Internet connectivity is not yet readily available.

3.2.1 *Challenges met by design*

The object-based fast prototyping approach we used (simplifying on Guerrero and Fuller, 1998) in developing databases and other aspects of the website (like the dynamic menus), addressed with various degrees of success each of these limitations. In most of the current web applications we looked at, the *server-mainly* approach had been preferred.[10] Such an approach gives the developer better control over what clients see and it reduces the resources necessary for developing and maintaining the application. However, it results in very long waiting times, low scalability and more hardware resources necessary at the server side, due to the fact that the entire interface (layouts and content) has to be built and transmitted to the client each time the client performs any action in the application. Since most of our target clients (Cree School Board employees and Cree community computers) are theoretically maintained by the same group of people, we chose a *balanced client–server* approach that allows us to decide the amount of work performed at the server-side and to reduce the amount of data transferred to the clients.

For example, the database interface is handled by one JavaScript file (generator) plus one more for each language supported by the database. These files, like the rest of the graphical elements of the interface, are loaded only once per session (under the default settings on all web browsers we know of). The language files are also dynamically generated each time a maintainer changes something in the parts of the interface to which the maintainers wanted access (Fig. 5). All the server sends to the client is a set of properties for the main JavaScript object (the interface generator), as for example an array containing the records requested by the user at any given time—in a very compact format.

The downside of this approach is that the various platforms and browsers that are being used require special attention. There are differences in the JavaScript support and object models implemented in each version of each type of browser, in the way screen measurements are done, and in the way multilingual text is supported. Therefore, the software that drives the interface has to be designed with these differences in mind. Since it is unreasonable to implement each (known) difference from the start, we develop for the small range of browsers and platforms that our intended users have access to, and from time to time (mostly when we receive bug reports), we modify the layout objects to accommodate more of them. Currently, we support Internet Explorer 6, and Firefox 1.5 on the Windows platforms. Macintosh systems which are Unicode-compatible are also supported (fully from OSX 10.3 upwards).

3.2.2 *The writing system*

You can see in Fig. 2 an example of Cree *syllabic text* as part of the oral record database. The Cree syllabic character set contains about 136 characters. No operating system supports this character set natively, but there are several font sets developed by researchers, publishers or enthusiasts in various formats, encodings and typefaces. Legacy operating systems like Windows 98 and lower, and MacOS 9.x and lower support only 8 bit fonts, which do not allow mixing of syllabic and roman characters in the same text field without a lot of overhead in the form of font formatting tags of some sort. Unicode fonts solve that problem, but: (1) they are not supported well on the legacy platforms mentioned earlier, (2) many of the Web programming tools available handle Unicode poorly if at all, and (3) the protocols define a lot of encodings, many of them

very verbose (up to a maximum of 7 bytes for each character). This limitation is currently overcome only on Windows and OS X platforms for most the DHTML interfaces, and on platforms supported by the Flash Player for the Flash interface.

3.2.3 *Making it snappy*

Finally, the limitation over which we have most control is *long transfer times*. As we mentioned before, we are separating content and layout by using Cascading Style Sheets and JavaScript objects in modular structures. We send very little redundant data from the server. Since the interface is programmed at the client side, we connect to the server less often and the interface is very responsive, without being overly crowded. There is a caveat here: hiding functionality behind buttons which redraw the interface has proved to be a problem for beginners, who did not know how to find what they were looking for, if given too much control over the searching and browsing process. We opted to have two versions of the interface, one for quick searching and browsing the entire catalogue (seen in Fig. 1), and one with category-based browsing and searching on specific fields (seen in Fig. 2). Normally, on the web, such a switch between a simple interface and a more complex one is achieved by making a new request to the server; our interface does not do that, since it renders either interface as chosen by the user. This increases the perceived responsiveness of the site and helps lower the strain on the network and on the server. The final result is that more users receive faster access to the resources offered by the same server.

3.2.4 *Choice of technologies—flash wins*

The conditional coding required to render an interface for such a heterogeneous set of clients is rather difficult to maintain, especially for visually oriented people. Our original choice of client-side support (HTML, JavaScript, and CSS), was driven by the following facts: (1) they generate a flowing, flexible layout, (2) they have an already large installed base, and (3) there is a relatively low cost-of-ownership of the tools required for development. However, since they require a programmer to check all changes, the cost-of-operation of the solution is higher than expected. As a result, we are looking at an alternative. We have already prepared several prototypes using Macromedia Flash (the widely used multimedia editing program which allows for web-embedded animation, sound and video streaming and lately, client–server applications). Versions before Flash MX2004 barely allowed for data transfer from the server and their Unicode support was poor. However, with this new version, these problems seem to be fixed and the only deterrents in using this visually oriented design and programming environment are (1) learning the Flash development style (peculiar for people new to multimedia) of half-visual-design, half-programming, (2) the higher initial investment price, and (3) the slightly higher difficulty of packaging Flash applications (originally designed for copyright protection), as open source, or even collaborative projects. None of these are insurmountable.

All the measures we discussed until now to reduce bandwidth[11] and memory footprint, have to do with the interface (buttons and other active *layout elements*, layouts) and the meta-data (in this case, *the description* of the oral records).

3.2.5 *Shrinking the sound*

As far as optimizing the content itself is concerned, anyone who has used digitized sound knows the large amount of storage needed for preserving sound in lossless formats. The sound format and sound-editing strategy of the oral material we chose resulted from our desire to balance speed, availability and protection from appropriation, with the perceived quality of the sound. The *mp3* format was clearly becoming the most widely used standard for web distribution of sound files, and since our project is not commercial, no licensing fees needed to be paid. After several tests with compressed Windows (wav), QuickTime (mov), various codecs for these formats, and the mp3-contender (ogg), we settled on an mp3 compression format. We judged that this format gives a sufficient quality for the use intended, i.e. web listening on the web and private CD burning, while preventing reappropriation of the material for commercial use and avoiding the

high-pitched artifacts of equivalent .ogg compression. In order to reduce the amount of time spent waiting for the sound to download, we had to cut our stories into parts. Some oral records are more than 30 min long, but we allow the casual listener to get a short introduction from which they can gauge the sound quality and make other aesthetic judgments about the voice and attitude of the storyteller before having to download the entire record. For the same reason, we split files larger than 2 MB in parts, and so, people with slow Internet connections can spread the download time over several sessions without the use of dedicated downloading software. After consulting our collaborators about the acceptable quality loss in the spoken story-like records, we opted for a sound compression ratio of 29.4 : 1 (sound digitized with a sample rate of 44.1 kHz and mp3-streamed at a rate of 24 kbps), or 22.1 : 1 (mp3-streamed at 32 kbps). That compression plus the 2 MB arbitrary file size limit we set, suggested a maximum of 10 min of content for any given parts, or a maximum wait of about 20 s per part on a fast connection (at average latency on a 512 kbps DSL), or about 2.5 min on a 56 kbps phone modem. There are issues of usability both for end-users and for maintainers.

We designed the sound player object in order to make use of whatever browser plug-in each computer has installed. The plug-in we are suggesting is Quick Time or Windows Media Player, since they appear as part of the interface and allow the user to control the sound (start/stop, volume, play-head location, etc.). However, if the user has installed some other sound player or sound-editing program that is set to handle mp3 files by default (like Winamp), our current solution opens that program as a helper in the background, with disconcerting effects (mainly, the database interface loses control over the playing sound, thus fails to terminate it when necessary). So we are considering writing our own player, in Flash, as we have done for the read-along and sing-along lessons (http://lessons.eastcree.org/).

3.2.6 *Conclusion*

To sum up, since we included user feedback in the core of the development cycle, we had generally good results on the programming side. The pattern language used assured consistency of data and metadata presentation while adding a level of layout flexibility almost impossible with classical database development tools. On the user side, the interfaces were very responsive even over 56 k modem connections, which resulted in many reports of user satisfaction. The pattern language implementation also allowed for implementation of additions or modifications suggested by users, resulting in increased motivation on their part.

4 Impact

The impact of the project on its intended audience cannot be quantified[12] easily. However, we can describe the feedback received from nonaffiliated end-users, from data maintainers, and other members of the Cree School Board, from Cree communities and the general Internet public.

4.1 The numbers (quantitative data)

Based on usage statistics over 6 months of server logs, we can get an idea of how the website is being used and by whom. We have complete data from October 2004 to March 2005, inclusive. Figure 7 shows hits to the site (chart on the left), the bandwidth load to the server (bottom-right chart, in red) and who is using the site. The latter is done by counting the identification numbers (IP numbers) of computers connecting to our site (this is called 'sites' and appears in orange in the chart on the top-right) and by counting visits with no interruption greater than 30 min ('visits', in yellow).

Total hits (in green) are not very representative since they include a variety of errors and hits resulting from web indexing engines like Google and Yahoo.

Files (in blue) represent the pages with their supporting content, like images, style sheets and JavaScript code files, minus the errors and easily identifiable machine-generated hits, but they are still not very indicative other than a general indication of usage over time.

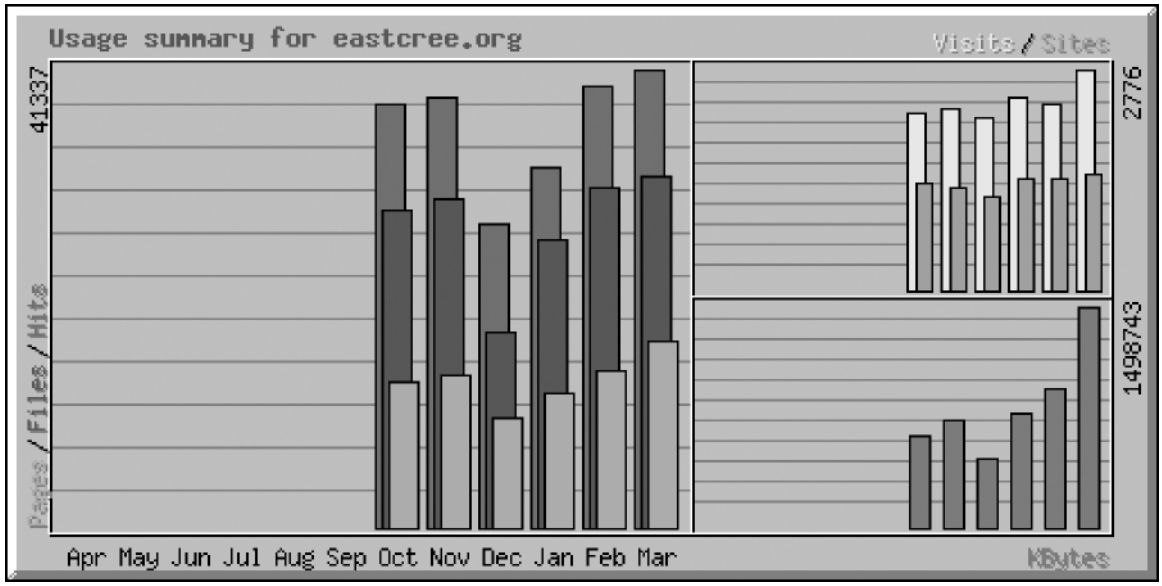

Fig. 7 Usage summary for eastcree.org (Oct 2004–March 2005)

Table 1 Summary by month eastcree.org (Oct 2004–March 2005)

Summary by Month										
Month	**Daily average**				**Monthly totals**					
	Hits	**Files**	**Pages**	**Visits**	**Sites**	**KBytes**	**Visits**	**Pages**	**Files**	**Hits**
Mar 2005	1333	1024	540	89	1470	1498743	2776	16756	31746	41337
Feb 2005	1420	1095	508	83	1403	942739	2344	14225	30685	39778
Jan 2005	1047	837	389	77	1395	770501	2412	12088	25970	32479
Dec 2004	886	565	319	70	1173	465306	2180	9909	17544	27476
Nov 2004	1291	984	455	76	1300	725507	2294	13655	29527	38758
Oct 2004	1228	921	422	71	1342	622859	2230	13110	28559	38097
Totals						**5025655**	**14236**	**79743**	**164031**	**217925**

Pages (in light blue) are much more reliable since they represent the actual content of the site (*html, swf*, sound files-*mp3* and downloadable documents—*pdf*)—presumably the reason why our intended users would use a site seeking information about (and in) the Cree language.

Table 1 gives the exact numbers for monthly and daily averages.

Since site testing is done primarily on our development machines, the numbers shown here are not biased in any strong way by our own browsing. Just to be safe, though, all numbers presented do not include any hits coming from a Carleton University address or from the IP number dedicated to the programmer's own machine.

The data presented earlier shows that the site usage is fairly consistent over the 6 months, with a trend towards a slight increase in usage reflected in all the statistics. This is shown in the monthly totals as well as in the daily averages.

Figure 8 shows which areas of the site attracted most interest, based on a count of files

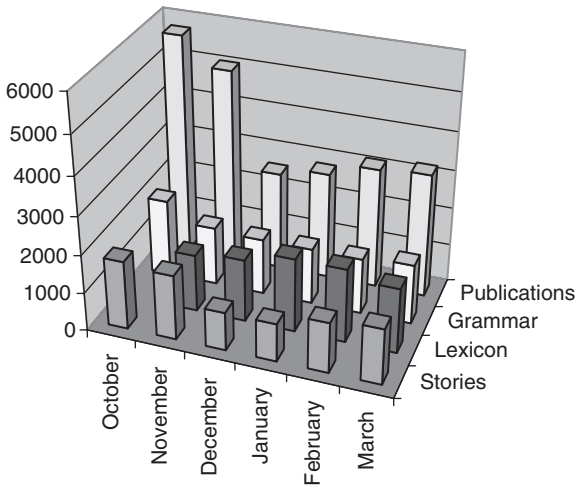

Fig. 8 Files per main areas over the reviewed duration

Table 2 Numbers of search strings from search engines which were followed to eastcree.org pages (only for January–March 2005)

Search strings related to.	Unique	Searches
Language	166	426
Cree	59	145
Technology	94	113
Other	29	37
All	312	667

per areas: *Publications* is consistently at the top. This is the first searchable catalogue of all the Cree School Board educational material we developed, and it has been on the site since 2002. The *Lexicon*, after it was released in November 2004 consistently ranks second. The *Grammar* pages come third. In fourth position is the *Stories* database.

For a more detailed analysis, we can start by looking at Table 2, which lists what the people who came to eastcree.org from search engines, were looking for. Under *Unique*, we list how many different strings people searched for, and under *Searches* we list the total number of searches in a given category. Most people had a language-related interest. The next largest category is Cree-related search strings. Many people came to the Resources section seeking answers to their technology-oriented questions on topics like Unicode encoding. The search string which brought the largest number of visits was 'southern dialect', with 139 instances. We will note that some search strings were included in more than one category, so the numbers under *All* are larger than the totals for their columns.

Examining the referrer information,[13] people have come to our site mainly from three sources: bookmarks on their computers, search engines, and two related East Cree sites: creeculture.ca and the Cree School Board site (http://www.cscree.qc.ca). This shows that the site is not yet linked from many places in the Internet, thus the activity the site has seen is from people actually interested in the topics the site addresses, rather than active promotion.

Tracing back the IP numbers recorded in the logs shows us that the vast majority of visits are coming from within the Cree School Board network (our intended primary target).

The site is now well-known by educators in all nine Cree communities since Cree Programs uses it for distributing updated teaching material in *pdf* format like their Spelling Manual, their Resource Book, etc. Table 3 shows us the most popular downloads over 6 months (Oct 2004–March 2005). The Resource book for teachers leads with 550 downloads per dialect, followed by the Spelling Manual (just under 500 downloads per dialect) and explanations about the Cree typing tools (429). Since Cree teachers and speakers typically use these, we can conclude again that we are reaching our intended audience. As for the interest for the Typing Tools explanations, it reflects an interest from technical people hopefully working on making the Web Cree friendly.

Some of the files are not linked within the site, they are only posted there for people to download when they are given the link. For example, the Medical terminology files were downloaded in March following a Medical Terminology workshop given by Prof. Junker and MacKenzie in February 2005 (see report by Bonspiel, 2005).

The ranking by average matches the ranking by total number of downloads.[14] This confirms our previous observation that the site is used very consistently, as opposed to just being randomly found by web surfers.

Table 3 Most popular downloads (Oct 2004–March 2005)

	Oct	Nov	Dec	Jan	Feb	Mar	Totals	6 month-Averages
ResourceBook-N.pdf	94	210	54	31	52	110	**551**	91.83
ResourceBook-S.pdf	64	100	129	56	83	118	**550**	91.67
SpellingManual_CreeNorthern.pdf	76	88	78	68	99	88	**497**	82.83
SpellingManual_CreeSouthern.pdf	68	62	104	41	80	86	**441**	73.50
CreeKeysUni.pdf	71	75	48	50	86	99	**429**	71.50
ECwebmap1.pdf	68	49	61	59	70	80	**387**	64.50
CreeConvManualCol.pdf	63	40	64	15	49	129	**360**	60.00
CreeConvManualBW.pdf	54	57	41	34	37	81	**304**	50.67
EastCreeNorthern-LexiqueProSetup.exe	2	6	44	73	80	64	**269**	44.83
EastCreeSouthern-LexiqueProSetup.exe	2	97	28	56	45	41	**269**	44.83
Unicode_Strategy.pdf	31	55	50	26	30	43	**235**	39.17
FAQ.pdf	25	28	40	39	36	30	**198**	33.00
DictOrderForm.pdf	n.a	52	25	31	36	38	**182**	36.40
CreeKeysPro.exe	34	32	20	22	30	38	**176**	29.33
flyer-bilingual.pdf	16	20	26	21	26	33	**142**	23.67
florrie-flyer-bilingual.pdf	18	20	21	14	19	30	**122**	20.33
orderform-bilingual.pdf	10	13	23	13	19	35	**113**	18.83
ManualCover.pdf	16	12	20	7	11	33	**99**	16.50
FMS-order-form-bilingual.pdf	13	10	22	8	16	22	**91**	15.17
Medical-Contagious.pdf	n.a.	n.a.	n.a.	n.a.	n.a.	77	**77**	77.00
TalkingSyllChart.zip	14	8	2	7	14	13	**58**	9.67
creeFonts.zip	6	7	3	7	6	14	**43**	7.17
Medical-Diabetes.pdf	n.a.	n.a.	n.a.	n.a.	n.a.	40	**40**	40.00
Totals:	**745**	**1041**	**903**	**678**	**924**	**1342**	**5633**	

4.2 The impressions (qualitative data)

In addition to site statistics, the impact the site is having can be somewhat measured by the e-mails received about it. Since the project started in 2001, Prof. Junker has been contacted twice per month on average by (other) aboriginal language speakers, educators and linguists, sending compliments, feedback and queries about the site. The most common questions are whether there is anything like that for another dialect/language they speak, teach or study, whether permission can be obtained for adapting the material and the site design to other dialects, and general comments about the site, including suggestions for improvements. Finally, there are questions from linguists using the site for mining data for linguistic research. Unlike other archiving practices recommended for endangered languages such as the e-meld initiative (http://emeld.org), the eastcree.org site was not primarily intended as an archiving and research tool for linguists. It can nevertheless be used as such (see Junker, 2005 for a discussion).

Another sign of positive impact is our involvement in the recent transfer of our open source developments to two other projects: the oral stories database engine is being used for a mapping project of oral history about places (Cree Regional Authority, creeculture.ca website) and the dictionary database is being adapted for a pan-innu dictionary project (MacKenzie et al., 2005).

We expect traffic to the site to increase as all teacher and Cree literacy training sessions by the Cree School Board now include an introduction to the site and on how to use its resources.

5 Conclusion

The creation of digital, online resources for threatened aboriginal languages presents many technical challenges. These challenges can only be met, if framed in the larger context of a research that also includes educational and ethical challenges. The form of these challenges will vary from place

to place, from one culture and language to another, but the general thread of adopting a participatory action research approach will help overcome them. The success in meeting our goal results from always keeping in mind our greater question: 'How can information technology help language preservation and documentation and how can the process of creating this resource have a positive impact on the language and its speakers?' The technical approach discussed here is thus framed as an answer to this bigger question. Fast-prototyping, open-source development, proprietary solutions, and database engine design choices were not explored on their own, they came as possible answers to our social and human concerns about the preservation of language and cultural diversity.

Acknowledgements

Research for this paper was made possible through the generous support of a research grant from the Social Sciences and Humanities Research Council of Canada and a grant for training workshops from the Ministry of Education of Quebec. We also wish to acknowledge the continuous support of the Cree School Board. Thanks to the many participants in our workshops and in particular to Frances Visitor.

References

2001–2005. Electronic metastructure for endangered languages data. US National Science Foundation Project. http://emeld.org

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., and Angel, S. (1977). *A Pattern Language*. Oxford: Oxford University Press.

Bonspiel, S. (2005). How do you say pancreas in Cree? School board and linguists collaborate to Cree-ate new words. *Nation*. Available at: http://www.carleton.ca/ecree/pdf/Nation-Terminology-03-05.pdf (accessed 18 March 2005).

Burnaby, B. (2002). How Have Aboriginal North Americans Responded to Writing Systems in Their Own Languages? Paper given at *the Atlantic Provinces Linguistics Association Conference*. St. John's: Memorial University.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Guerrero, L. A. and Fuller, D. A. (1998). *Objects for Fast Prototyping of Collaborative Applications. Proceedings of the 4th CYTED-RITOS International Workshop on Groupware, CRIWG'98*, Rio de Janeiro, Brazil, September, 1998.

Jancewicz, B. and Junker, M.-O. (2002). Cree on the Internet: How to Integrate Syllabics with Information Technology and the Web. Presented at the *34th Algonquian Conference*, Kingston: Queen's University.

Jancewicz, B. and Junker, M.-O. (2003). Frequently asked questions about Cree syllabics, computer technology and the web. In www.resources.eastcree.org, web pages and PDF download.

Junker, M.-O. (ed) (2000–2006). The East Cree Language Web. Available online at: www.eastcree.org.

Junker, M.-O. (2002). Participatory Action Research in Linguistics: What Does it Mean?/La recherche participaction en linguistique: Enjeux et significations. Presented at the *Session on Ethics of Archiving Languages and Fieldwork, organized by the Aboriginal Language Committee*, *Canadian Linguistics Association Annual Congress*. Toronto: University of Toronto.

Junker, M.-O. (2005). The eastcree.org Project: Reflections on Participatory Action Research, Information Technology, and Tools for Linguistic Research. Presented at the *10th Workshop on Structure and Consituency in the Languages of the Americas (WSCLA 10)*. Toronto: University of Toronto.

Luchianov, M.-R. (2000). *MoStaCon: Usability Study for an Experiment Design Tool*. Master Thesis, Sofia: New Bulgarian University. Available online at: http://www.monicsoft.net/port/nbu/COG699.pdf.

MacKenzie, M., Branigan, P., Burnaby, B. and Junker, M.-O. (2005). Knowledge and Human Resources for Innu Language Development. Community-University Research Alliances Project. Funded by the Social Sciences and Humanities Research Council of Canada (2005–2010). http://www.innu-aimun.org/

Morris, M. and Muzychka, M. (2002). *Participatory Research and Action*. Ottawa: Canadian Research Institute on the Advancement of Women.

Spinellis, D. (2001). Notable pattern languages for domain specific languages. *Journal of Systems and Software*, **56**(1): 91–99.

White, I. (1994). *Rational Rose Essentials: Using the Booch Method*. Redwood City, CA: Pearson Benjamin Cumming.

Zhang, J., Chang, C. K. and Chung, J.-Y. (2003). 'Mockup-driven Fast Prototyping Methodology for Web Requirements Engineering', *Proceedings of the IEEE 27th Annual International Computer Software and Applications Conference* (COMPSAC 2003), November 3–6, 2003, Dallas, TX, USA, pp. 264–269.

Notes

1 Layouts are ways to place different graphical elements (text, images, separators, colors), on a canvas or screen, relatively to each other. In the publication catalogue, as well as in the oral database, we developed an innovative way to generate and display pages. The layouts of the maintainer's interfaces are still not multilingual, and it is still rather difficult to edit interface terminology out of its context. The next phases of our research will focus on integrating the various interfaces into one layout that will be used both for browsing and editing, depending on who the current user is and what action the user chooses at a given moment. Steps toward this have been taken in the interface to the terminology forum (http://ayimuwin.eastcree.org/). These steps have become much more accessible due to the recent improvements in Flash technology, as described in the Section 3.2.4.

2 A CMS or content-management system is a set of tools used to organize and facilitate collaborative content creation. A distributed CMS allows people living far away from each other to work on the same content. 'Distributed', in this case, refers to the location of the users. It is possible to distribute the content too (to allow for separate repositories e.g. for reasons of copyright, ease of editing, storage safety), but in our case the content is still centralized: it resides on a single server from where everyone retrieves it.

3 http://en.wikipedia.org/wiki/List_of_content_management_systems

4 This approach reduces the task of adding a new language or dialect to a matter of adding and populating one column in two of the database tables, changing the content of a few rows in other tables—and of course some relatively minor debugging—all of which can be done directly online. However, as for most web application development, whenever we make major changes to the database we do it safely on a development server and we upload changes to the public server only after preliminary debugging.

5 In short, the relevant steps of the entire loop of analysis (separation and integration) is defined at the same time. Unlike XML and its derivatives (e.g. COM, DTD, and XML schemas for semantics, XML, XSL/XSLT for content/syntax, XSLT for style), which attempt the same separation, MonDoc suggests using standards only as points of departure rather than as the entire model. The primary pattern of MonDoc is that the content and both its semantics (object definitions) and its style definitions, are to be developed together rather than separately, and in the context of the document itself, rather than solely as separate documents and applications.

6 'Skins' are names usually given to layout templates which can render the same content (data and metadata), in different ways, sometimes at the flick of an option toggle. See for example http://www.csszengarden.com/

7 Markup is a name given to the format of the metadata describing a piece of text. Some metadata can enclose other metadata, thus describing in more and more detail the enclosed data. For example in text, the text is first made bold then italics.

8 CSS or 'Cascading Style Sheets' is a web technology that allows for separation of layout and content. However, different web browsers support different subsets of the technology, which makes using CSS less powerful than it was intended and therefore requires additional support through providing different page content depending on the browser used at any given time; this can be done at the server or dynamically in the browser.

9 Fast-prototyping is a methodology that suggests that drafts, quick sketches or barely functional versions of each feature to include in a project have to be checked by designers and preferably by end-users before continuing production. This is based on the observation that features look different in descriptions as they do when actually used; different by themselves than when used in conjunction with other features of the same system. So, many times these feature drafts have to be scrapped or tabled for later reorganization. The word 'fast' in fast-prototyping refers to releasing/checking test versions very often, not to the total speed of development. In fact, in cases where the feedback loop between developers and users/designers is slow, it can take much longer than when using other methods. The main advantage, though, is better fit of the prototypes and final products to the task targeted and to the intended user base.

10 Dreamweaver and FrontPage templates, Syndeo, WebCT, Wikis, specialized open source catalogues.
11 Bandwidth is used to describe both the *available data rate* between a server and client (in multiples of bits(b) or bytes(B) per time unit, e.g. kbps or GB/month) and *actual transfer* from the server (in multiples of bytes, e.g. Kb or MB). It is desirable to have more of the former and less of the latter. MonDoc aims at dramatically reducing the latter.
12 For this surface analysis of the server logs, we used the Open Source tool http://webalizer.org and scripts written by Luchian for a few simple data-mining tasks.
13 The referrer information is a record of the previous sites people were on before they followed a link to our site.
14 The only exception to this are the files which were not available over the 6 month period.