Ney Yibeogo - Hello World: A Voice Service Development Platform to Bridge the Web's Digital Divide

André Baart[1], Anna Bon[2], Victor de Boer[3], Wendelien Tuijp[4] and Hans Akkermans[5]

[1]*Amsterdam Business School, Universiteit van Amsterdam, Plantage Muidergracht 12,*
1018 TV, Amsterdam, The Netherlands
[2]*Network Institute, Vrije Universiteit Amsterdam, De Boelelaan 1081, 1081 HV, Amsterdam, The Netherlands*
[3]*Computer Science Department, Vrije Universiteit Amsterdam, De Boelelaan 1081,*
1081 HV, Amsterdam, The Netherlands
[4]*Centre for International Cooperation, Vrije Universiteit Amsterdam, De Boelelaan 1081,*
1081 HV, Amsterdam, The Netherlands
[5]*Network Institute, Vrije Universiteit Amsterdam, De Boelelaan 1081, 1081 HV, Amsterdam, The Netherlands*
a.h.j.baart@uva.nl, {a.bon, v.de.boer, w.tuijp}@vu.nl, hans.akkermans@akmc.nl

Keywords: Digital Divide, Low Literacy, Sub-Saharan Africa, Voice-based Services, Low-resource Hardware, Services Development Software Kit.

Abstract: The World Wide Web is a crucial open public space for knowledge sharing, content creation and application service provisioning for billions on this planet. Although it has a global reach, still more than three billion people do not have access to the Web, the majority of whom live in the Global South, often in rural regions, under low-resource conditions and with poor infrastructure. However, the need for knowledge sharing, content creation and application service provisioning is no less on the other side of this Digital Divide. In this paper we describe the *Kasadaka* platform that supports easy creation of local-content and voice-based information services, targeted at currently 'unconnected' populations and matching the associated resource and infrastructural requirements. The Kasadaka platform and especially its Voice Service Development Kit supports the formation of an ecosystem of decentralized voice-based information services that serve local populations and communities. This is, in fact, very much analogous to the services and functionalities offered by the Web, but in regions where Internet and Web are absent and will continue to be for the foreseeable future.

1 INTRODUCTION

The World Wide Web is a unique public space for knowledge sharing, content creation and application service provisioning for billions on this planet. Although it has a global reach, still more than three billion people do not have access to the Web: the 'Digital Divide' (Fuchs and Horak, 2008). The majority lives in the Global South, often in remote rural regions, under low-resource conditions and with poor or even absent infrastructures.

However, needs for knowledge sharing, locally relevant content and application service provisioning are certainly no less beyond the current borders of the Web.

To overcome the Digital Divide, various policies are promoted to improve global access to Internet, Web and its vast arsenal of resources. A prominent one, for which quite large funds have been made available by donors such as the World Bank, is the attempt to roll out forms of "affordable Internet" to currently unconnected regions.[1] Basically, the underlying idea is a form of relatively straightforward technology transfer from advanced countries to developing and emerging regions (USAID, 2017; Schmida et al., 2017; The World Bank Group, 2016).

Our research focuses on information exchange and knowledge sharing support for small holder and family farmers in the African Sahel (including e.g. Mali, Burkina Faso, northern Ghana). In a country such as Mali, around 80% of the population depend for their livelihood on work in small subsistence agriculture in remote rural regions where there is no Internet, very limited electricity, and high levels of low-literacy (around 50% on average, for women even significantly higher). Under these conditions it is highly unlikely that a technology transfer policy of Inter-

---

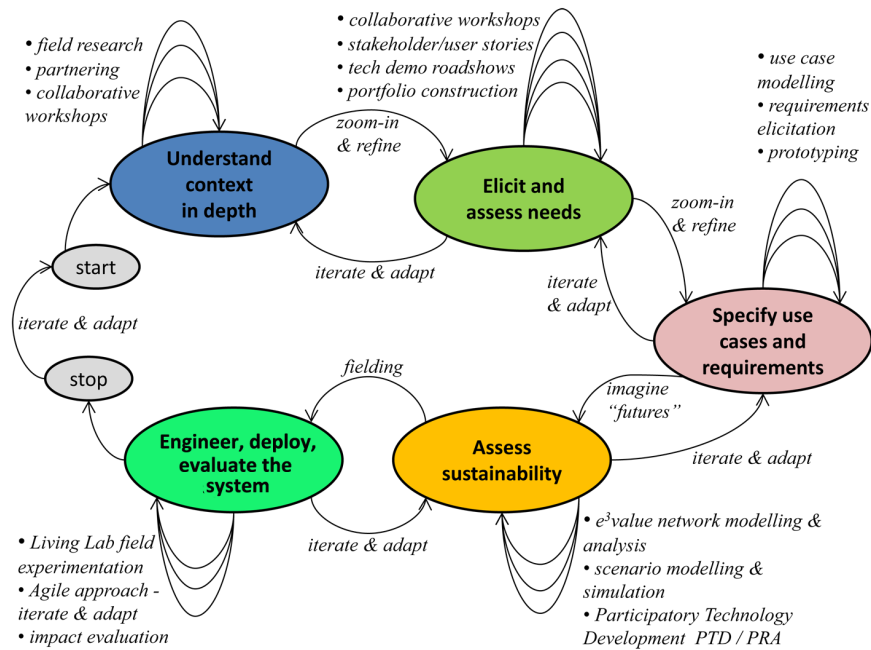[1]See: https://webfoundation.org/our-work/projects/alliance-for-affordable-internet/

Figure 1: ICT4D Field research methodology, from (Bon et al., 2016).

net roll-out to bridge the Digital Divide will come to fruition in some foreseeable future.

This does not imply that nothing can be done. The contribution of this paper is that one can, and that it is possible to develop and deliver web-reminescent services for information and knowledge exchange, but not in a one-size-fits-all technology transfer approach. It requires a thorough investigation *in the field* of conditions, requirements and local specificities. This leads to insights and technical directions that cannot be derived from advanced but far-away technology considerations alone.

Accordingly, in this paper we present the *Kasadaka* platform intended to support easy creation of local-content and voice-based information services, targeted at currently 'unconnected' populations and meeting the harsh conditions at the other side of the Digital Divide.

The Kasadaka platform and especially its Voice Service Development Kit aims to facilitate the formation of an ecosystem of many decentralized voice-based information services that serve local populations and communities. This is, in fact, very much analogous to the services and functionalities offered by the Web, but in regions where Internet and Web are and will continue to be absent for the foreseeable future.

2 APPROACH AND METHODOLOGY

The design of a service development platform for low-tech and low-resource environments as sketched above, cannot be based on generic technological considerations alone, but requires in-depth study of local contexts and conditions, in addition to and beyond the usual ones that hold for ICT development in general.

In our approach, we cater for this situatedness of information systems and services by carrying out extensive field research. The underlying — iterative and collaborative — field research methodology is depicted in Figure 1 in the form of an intention-strategy map (Rolland, 2007), and is discussed in more detail in (Bon et al., 2016).

Subsequently, this field research is utilized to derive requirements for the platform (discussed in Section 3). Apart from a good understanding of the 'Digital Divide' context in which services are to be deployed, platform requirements originate from the spectrum of service use cases the platform is supposed to support and execute. These use cases, developed with end users and stakeholders in the field, similarly are the result of field research, often in the form of collaborative workshops.

Thus, the methodology employed to come to the *KasaDaka* service platform services for low-tech low-resource environments is in brief:

- Field-based context analysis: i.e. finding out the local conditions with respect to technical infrastructure, environmental conditions, availability of technical support etc;

- Platform requirements analysis, as also derived from the service use cases, elicited from local users during field research;

- Technical design and implementation of the platform;

- Testing and evaluation, also with local users in the local context.

Below, Section 3 presents an analysis of the platform requirements, Section 4 describes the architecture and technical implementation issues, and Section 5 describes various evaluations of the platform. Sec. 6 discusses related work and Section 7 summarizes our main conclusions.

3 PLATFORM REQUIREMENTS ANALYSIS

Requirements in a technology development project are derived from the needs that the context of the project brings to the table. The development of a system that is intended for those on the other side of the Digital Divide has to deal with several combinations of circumstances and issues that are rarely encountered in technology development projects in the developed world.

3.1 Societal Challenges

Voice-based access to information is an essential requirement for bridging the digital divide, and reaching the world's rural poor. In these populations, literacy rates are low, which disqualifies any service that is text-based. In several sub-Saharan African countries (such as Niger, Mali and Burkina Faso) the literacy rates are below 40%, which makes the vast amounts of textual information on the Internet out of reach for a major part of the population in these regions (UNESCO, 2011). Furthermore, many indigenous cultures have a strong oral tradition in communication, so that voice-based services have a natural fit with the locally already existing means of communication.

Voice services for the world's rural poor have to support under-resourced languages, which implies that they can not use advanced speech technologies.

While many developing countries have a technologically well-supported official language (often a remnant of colonial times), this language is not necessarily spoken by the entire population. Rather, the local population speaks their own indigenous language which is tied to their local region. Africa for instance has around 2000 local languages, which each often have local dialects (Heine and Nurse, 2000). The majority of these languages are *spoken languages*, meaning that there exists little to no literature in these languages. Furthermore, due to the populations speaking these languages being poor and relatively small, these populations do not provide a profitable market for the development of Text To Speech, Automatic Speech Recognition and Natural Language Processing technologies in these languages. Most of the recently developed voice platforms that offer complex information services (e.g. Apple's Siri, Amazon Alexa, etc.) rely on the use of these technologies. While these technologies are in widespread use around the world, they require significant research and a lot of work (and thus significant financial investment) in order to support a language at a level that is sufficient for usage in voice services. (Bagshaw et al., 2011; Black and Lenzo, 2000; Farrugia, 2005; McTear et al., 2016; Besacier et al., 2014; de Vries et al., 2014) The number of languages that has well-developed speech technologies is rising, but these do (almost) not include any of the indigenous languages found in the developing world. This situation is not likely to change, as there is little (financial) incentive to develop technologies for these languages. Taking into account these restrictions, these languages are referred to as under-resourced languages (Berment, 2004).

3.2 Resource and Infrastructure Constraints

Information Services for the World's poor should be affordable and accessible through locally adopted technologies, i.e., especially mobile (dumb-)phones. Developing countries are some of the poorest in the world, where large parts of the population live on less than € 2 per day.[2] In order for a voice service to be of use to the general population, the cost of accessing and using it thus have to be very low. This implies that the users should be able to access the service without having to purchase a new device or service, but rather using a device they already own or have access to. The initial and running costs of a voice service should also be low enough to be affordable (and to provide sufficient return on investment) for

---

[2] Sub-Saharan Africa: https://data.worldbank.org/indicator/NY.GDP.PCAP.CD?locations=ZF

the rural poor.

The voice-service platform should function with limited infrastructure. The (digital) infrastructure in these countries is unreliable and expensive, especially in the rural areas. While some villages have access to electricity, it is often unreliable and black-outs often happen multiple times per day (also in cities). The majority of the population does not have (direct) access to electricity [3]. The hardware that hosts access to Internet is slowly becoming more common (Poushter, 2016), but is very expensive and unreliable[4], due to a lack of local hosting and limited international (backbone) connections. While Internet adoption is low, mobile phones have become a successful means of communication in much of the developed world, having become the main means of telecommunication in sub-Saharan Africa (GSMA, 2016). The coverage of mobile telephony networks is often quite good, covering a large part of the population (also rural areas).

3.3 Financial Sustainability of Services

The platform should be able to provide financially sustainable voice services. This is achieved by reducing the cost of voice services — which consist of hardware costs, development costs and maintenance costs — as far as possible. This has consequences for all elements in the architecture of the platform, which have to be chosen and designed in such a way that costs are minimized. Financial sustainability assures that the services are targeted at the needs of local communities and thus provide sufficient value to offset their cost.

The platform should facilitate development by local developers with limited programming skills. The development process of voice services should thus be simple, flexible, not require advanced programming skills and should take place in a graphical interface. A very small amount of the population owns (or has access to) a computer, let alone a connection to the Internet. Also, there are few local software developers and technicians available for the development and maintenance of local infrastructure, systems and applications. From this small pool the amount of software developers that have experience with voice services will thus likely be extremely low. Hiring foreign developers is not an option, as the cost of foreign labor is extremely high, conflicting with the above requirement of financial sustainability. Ensuring local development is thus essential for the

formation of a voice service ecosystem targeted at the unconnected. In order to increase the size of the potential pool of voice service developers, the process of development should be accessible to users that do not have programming skills. This simplification should allow for people with a basic understanding of using computers to be trained in the development of voice services. Besides the financial aspect of local voice service development, an additional benefit is that local developers have a smaller distance to the end-user of the voice service, not only in a spatial sense but also in social and cultural sense. This improves the local relevancy of the services as well as the understanding of the end-user's needs. The platform should facilitate the formation of small businesses and entrepreneurs that are specialized in developing and hosting voice services, enabling them to make a living from selling customized voice services to local companies and communities.

The platform should run on low-resource hardware and be based on Free/Libre and Open Source software. In order to keep the costs of running voice services as low as possible — and thus contribute to the financial sustainability — the hardware used in the platform should be cheap, robust, and consume little energy. Another aspect that influences the cost of the platform is the cost of software licenses. The prices of commercial telephony products and other software are at a level that is acceptable in the Global North, which is not affordable in the developing world. Furthermore the liberating nature of open-source software allows for the practice of *bricolage*: tinkering with existing technologies in new and innovative ways which allows for the formation of successful innovations (Ali and Bailur, 2007). Accepting that the usage of technology cannot be tightly controlled and that successful innovations often come from unexpected directions, can be a determining factor of success. By explicitly granting the general population the freedom to use the technology in any way they see fit, practicing *bricolage* is facilitated and the available technology is more likely to be (eventually) applied in a way that is most relevant and innovative to the local context.

3.4 Example Use Cases of Voice Services

Below we outline two examples of the types of voice services that the platform should facilitate. These use cases have been elicited and analyzed during our various field visits to Mali and Burkina Faso.

Foroba Blon, a system for village reporting. We briefly describe here the case of Radio Sikidolo, a

---

[3]https://data.worldbank.org/indicator/EG.ELC.ACCS.ZS?
end=2014&locations=ML&start=2014&view=map
[4]http://100mega.ml/

small radio station in Konobougou, a village in the south of Mali several hours from the Malian capital Bamako. It reaches up to 80,000 listeners in the region. According to its director, Adama Tessougué, this radio works with free-lance village reporters who collect news and announcements in the surrounding villages for broadcasting. Example topics are wedding announcements, funerals, lost animals, interviews and interesting stories. In the absence of Internet in these remote areas, village reporters use simple GSM mobile phones to send news to the radio. For this, the program maker at the radio station had to be available in person on the phone, and then write down the incoming information on paper for broadcasting. Evidently, this task is time consuming and inefficient. Foroba Blon is a voice-based system allowing village reporters to phone in and to submit spoken news items that are off line stored in the system (Gyan et al., 2013). Messages can then be accessed and managed by the radio journalist through a web interface on his laptop, without the need for Internet. The radio station uses the messages for interactive programming, or receives (financial) compensation for the spreading of advertisements and announcements. The Bambara name Foroba Blon refers to the Malian village square where everyone is allowed to speak out, though respectfully.

The Foroba Blon use case has been used during the evaluation of the platform, which is covered in Section 5.2

Weather information crowd-sourcing. Many farmers and families in Burkina Faso depend on rain-fed agriculture. The rainy season is short (three months) and so pertinent information on actual and forecast rainfall is extremely important, for example, to better plan cropping calendars and improve harvests. During recent collaborative use-case and requirements workshops in Gourcy, Burkina Faso, organized by local NGO Réseau MARP, regional radio stations, the association of innovative farmers in the Yatenga province, and the W4RA team of authors, it became abundantly clear that important weather information never reaches local farmers in Burkina Faso. Global weather information is in principle available through the Web, but it is not accessible to farmers that face the familiar issues of lack of electricity, of digital infrastructures, and issues of language and literacy. Furthermore this information is often inaccurate, due to a lack of measurement infrastructure and accurate weather models. The Burkina Faso weather voice service allows farmers to receive data on the amount of rainfall, as measured by fellow farmers that have a measurement bucket on their land. These farmers call

in their measurements periodically. Besides providing other farmers with essential information, the information is also used to accurately track historical rainfall in the region.

4 KASADAKA TECHNICAL IMPLEMENTATION

The platform that we propose is called *Kasadaka* (*talking box* in a number of northern Ghanaian languages). The platform consists of a combination of hardware and accompanying software. Figure 2 is a visual representation of the architecture of the system and highlights the interactions between the components.

4.1 Hardware

The hardware forming the foundation of the KasaDaka platform is the Raspberry Pi, which is a low-resource computer based on an ARM processor (like found in many smart phones). The main advantages of the Raspberry Pi are it's low power consumption (and subsequently no need for cooling), good on-board connectivity and the low price[5] (and thus also a low replacement cost). As the Raspberry Pi does not include a Real Time Clock (RTC), it cannot accurately keep time when the power is lost. To solve this problem, a small and cheap battery powered RTC is connected to the Pi's general connector. The Raspberry Pi is a very popular product for experimentation and many projects, and is thus widely available, making it easy to replace should hardware problems arise.

To provide the Raspberry Pi with connectivity to the local mobile phone network, a USB 3G modem is used. The exact make and model of this modem can differ, as long as it is on the supported hardware list[6] of the `chan_dongle` Asterisk extension.

4.2 Software

On top of the Raspbian Operating System several applications run that work together to provide the voice-service functionality. Almost all applications used are open-source and thus free to use and adapt.

Telephone exchange software: Asterisk. Asterisk is a very popular open-source Private Branch Exchange (PBX) telephony application. It is used

---

[5] A Raspberry Pi 3 (including case, power supply and SD card) costs around € 60 at the time of writing.
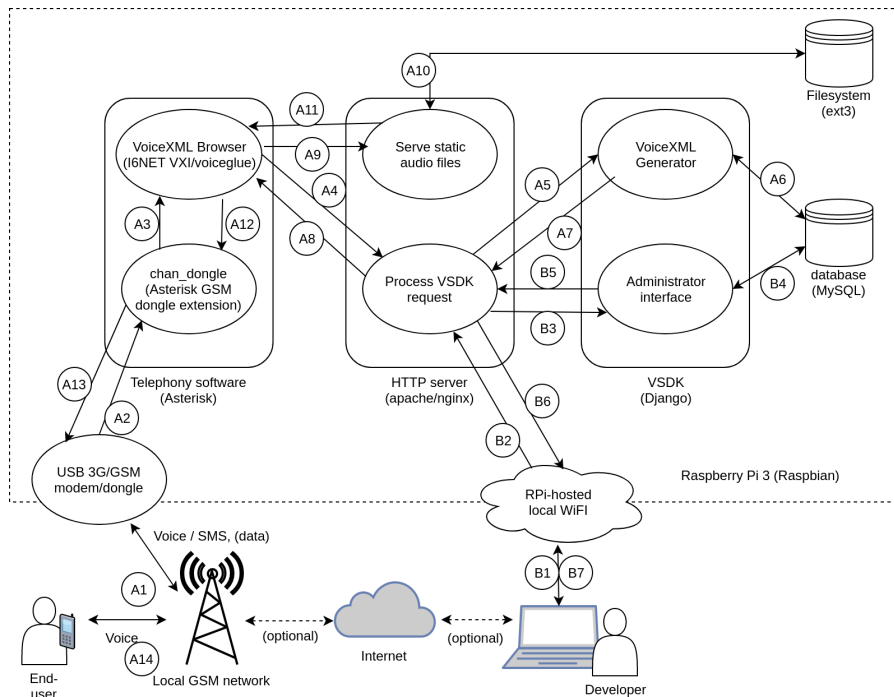[6] https://github.com/bg111/asterisk-chan-dongle/wiki/ Requirements-and-Limitations

Figure 2: Overview of the Kasadaka system architecture.

for the routing of incoming calls to its destination using Voice-over-IP technologies. In the use cases of the KasaDaka platform, Asterisk provides the connection between the phone network (3G dongle) and the VoiceXML interpreter. To enable Asterisk to interface with the 3G dongle an extension is required. Kasadaka uses `chan_dongle`[7], which is an open-source Asterisk extension that provides connectivity between GSM/3G modems and Asterisk. It enables Asterisk to receive and place calls using the connected modem, as well as send and receive SMS messages.

Voice application document standard: VoiceXML. VoiceXML[8] is a document standard for voice applications, based on XML. It is a standard designed by the World Wide Web Consortium and is used for creating documents that describe voice-based interactions. It supports interactive voice dialogues between the computer and the user and usually contains text (in written form) that is later processed by a TTS engine. Responses by the user can happen through pressing a number on the phones keypad of by speaking (for this ASR needs to be available). As the voice applications that use the Kasadaka framework mainly focus on under-resourced languages, TTS and ASR are not used (nor available). Fortunately VoiceXML

also supports the playback of audio files, much alike embedding images in an HTML page. This allows the use of pre-recorded fragments to build up the voice services, but restricts the way of interaction to using the phone's keypad. A VoiceXML document is 'rendered' for the user in a way that is comparable to the rendering of a HTML file in a web-browser, but in this case is done by a voice browser.

VoiceXML interpreter: VXI. The software component that is used for 'rendering' VoiceXML files is VXI[9], a closed-source VoiceXML interpreter built by the company I6NET[10]. VXI connects with Asterisk as an end-point for incoming calls. When a call is redirected to VoiceXML a pre-configured URL is passed on to VXI, which it loads and 'displays' to the user as initial voice interaction. Normally this is the principal document belonging to a voice service. VXI currently is the only closed-source component used in the KasaDaka platform. While the goal is to use only open-source software, there is no currently maintained open-source alternative.

HTTP server: Apache. VXI loads the VoiceXML files it interprets over a HTTP connection, just like loading a HTML page on the web, but locally. In order to serve these files (and the audio files that are

---

[7]https://github.com/bg111/asterisk-chan-dongle
[8]https://www.w3.org/TR/voicexml21/

[9]http://www.i6net.com/technology/voicexml-ivr/
[10]http://www.i6net.com/

referenced in the VoiceXML files), a web server is required. There are many open-source web-servers, one of the most used is Apache 2.

VSDK development framework: Django (Python). In order to make the VSDK easy to extend by developers, Python is a programming language of choice as it is a popular language that is well supported and has several popular web-frameworks. As VoiceXML documents are comparable to HTML documents, most web-frameworks can also be used to generate VoiceXML files. Django[11] was chosen as the Python web-framework, as it has very good and extensive documentation, is well-supported and follows a Model-View-Controller (MVC) methodology (Krasner et al., 1988). Django is open-source and has a rich collection of projects and libraries that can be used to extend it's functionality. Django has a good implementation of internationalization functionalities, which enable the interface of the administrator interface to be translated to different languages.

4.3 Voice Service Development Kit

The main goal of the Voice Service Development Kit[12] (VSDK) is to support the development of voice-services in the context of the developing world. As the voice services are hosted on a Raspberry Pi and Internet connectivity is not to be expected, the development of voice services happens off line. Using a web-based interface is preferable to running a development environment on a computer because it solves problems with compatibility (different devices, operating systems) and reduces complexity (does not require installation of software). Another advantage of this approach is that the development and hosting of voice services are integrated, allowing for instantaneous results (and testing) of changes made to the application. The VSDK is hosted on the Raspberry Pi, which also hosts a local wireless network, through which it is accessible. Local entrepreneurs can use the VSDK to develop voice applications on the Kasadaka platform, without requiring them to have programming skills.

The structure of the voice-application is stored in the database, using Django's model functionality. When an element in the voice-application is requested by the user in a phone call, the VoiceXML interpreter (VXI) requests the element through an HTTP call. Django then retrieves the information about this element from the database, and uses a view to 'render' the element in VoiceXML. The VoiceXML interpreter

---

[11] https://www.djangoproject.com/
[12] The VSDK's code can be found on GitHub. See: https://github.com/abaart/KasaDaka-VSDK

then interprets this VoiceXML file and 'displays' it to the user.

While the interactions in voice services are always different, most of them can be generalized to a small set of interaction types, such as making a choice, playing back an audio message, or recording (voice) input of the user. The VSDK provides a set of these *building-blocks*, which consist of a VoiceXML template, view and an administrator interface to use and customize them. The current set (which will be expanded in the future) consists of a menu-based interaction, recording of user voice input and the playback of messages. While this set is limited, it offers sufficient functionality for many voice services and serves as a demonstration of the method of voice service development.

Voice-services in the developing context have to support under-resourced languages, for which there are no speech technologies available. The VSDK supports different languages in voice services by utilizing pre-recorded audio fragments that are relevant for the use-case domain. During the development of the service, all the necessary voice-fragments are recorded in the different languages in which the service has to be accessible. These voice-fragments are stored in the file system and referenced in a "voice label" element that is stored in the database. This voice label refers to voice-fragments that represent a fragment of text, spoken in different languages(de Boer et al., 2015).

5 EVALUATIONS

The hardware configuration of the Kasadaka platform has already been tested previously in several pilot deployments. These deployments have shown that the hardware runs well in the conditions of sub-Saharan Africa and allows voice services to be hosted independently on all encountered mobile telephony networks.

The evaluation of the VSDK is structured in two steps: the first was an evaluation in the Netherlands with inexperienced users, the second validation was a case study in Mali with a user from the intended user group of the VSDK.

5.1 Evaluation by Development of Several Use Case Prototype Services

The VSDK was evaluated with 10 student groups during the ICT for Development (ICT4D) course at the Vrije Universiteit Amsterdam. The groups each developed a voice service for several distinct use cases, which were co-created with rural communities and relevant in the context of the developing world. The
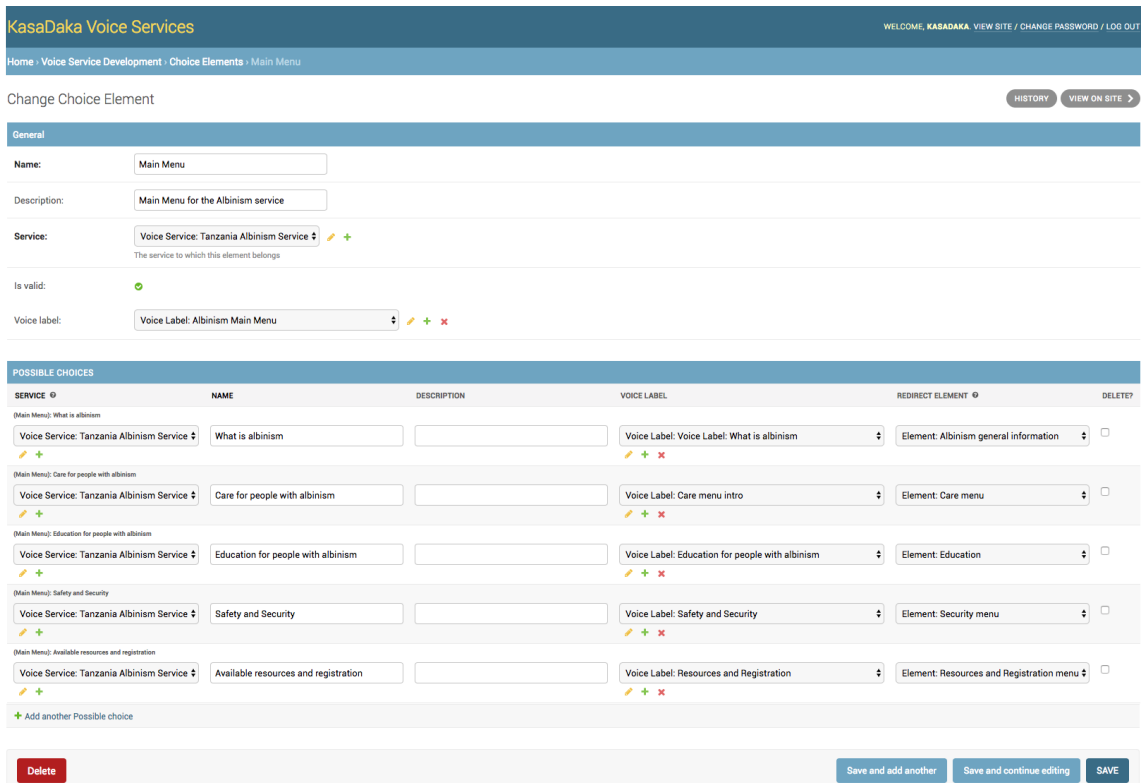
Figure 3: An example screen shot of the voice-service development interface of the VSDK. Shown is an example of a choice interaction element. A voice service developer uses this GUI to develop voice-based applications on the Kasadaka platform.

Figure 4: Adama Tessougué of Radio Sikidolo shows the Kasadaka on which the Foroba Blon voice service now runs with its Bambara language interface.

Figure 5: André Baart and Adama Tessougué evaluating the VSDK running on the Kasadaka platform, at Radio Sikidolo in Mali.

choice to evaluate with students was made because of the ease of communication with the students, which is significantly less complex and expensive than travelling to a developing country. While the level of computer literacy of the students is higher than that of the intended voice service developers in developing countries and the evaluation took place in the Netherlands, feedback of the students is still very relevant

for verifying the concept of the VSDK. The VSDK proved to be successful providing the required functionality for the creation of basic voice-service services that can be used for rapid-prototyping purposes. Using a graphical interface, voice-services consisting of simple choices with associated options and messages can be designed without having to write any code. These prototypes can be made in a quickly and without extensive knowledge of the underlying

technologies, which is useful for rapid prototype development and evaluation; After set-up, a simple service can be developed and tested in under 30 minutes, however the development of complex use-cases takes more time. During the course, 80% of the student groups had successfully built a working voice service using the VSDK. These 8 applications were developed for 5 distinct use cases. The included interaction templates allowed the students to quickly build demonstration prototypes of their voice services. In order to provide more complex functionality in their voice services, 78% of the student groups had extended the functionality of the VSDK with data models specific to their use case and 67% of the groups extended the VSDK with additional interaction templates.

At the end of the course the students were asked to fill in a short survey on their experience with creating a voice service and using the VSDK. The goal of this survey is to learn about the process that the students went through as they developed their first voice service. The survey consisted of statements about the usefulness of the VSDK, which had to be answered in a Likert scale. There were also qualitative questions about VSDK features, improvements and suggestions, as well as questions about their perceptions during the development process.

This evaluation has shown that the methodology of building-blocks that is used in the VSDK allows for the development of simple voice services by inexperienced users, which was the goal. It also provided insight in the limitations and problems of the VSDK. The main limitation lies in the are of user generated data management. The VSDK does not yet allow the creation of custom data models from the development interface. Other limitations were the limited set of user interactions provided and the impossibility of the integration of external data sources. These limitations prevent the VSDK of being suitable for more complex voice-services, as 'traditional' voice-service development skills are still required. In the case of a custom extension to the VSDK, the functionality of this extension can be reused throughout the application and shared with the rest of the development community (through GitHub). Furthermore the administrator interface can easily utilized by these custom extensions, which allow voice-service maintainers (without programming knowledge) to change settings and other elements of the extension's functionality. Thus after the development of the extension is completed, maintenance can still be performed by others without knowledge of it's inner workings, maintaining the advantage of ease of use offered by the VSDK.

5.2 Case Study: Radio Sikidolo

The results and knowledge of the first evaluation have been used in the second iteration of the VSDK, which has been evaluated in collaboration with Adama Tessougué, the director of Radio Sikidolo in Konobougou, Mali. For more information see Section 3.4 While the evaluation with the students was sufficient for a general validation of the methodology of the VSDK, it did not evaluate the VSDK while running on the hardware of the Kasadaka platform, and was not in the intended context of a developing country. This evaluation addresses these limitations: it evaluates the VSDK and the Kasadaka hardware and software as a whole, in the intended developing world context. This validation session was done at Radio Sikidolo, which has electricity and a relatively stable Internet connection, the latter of which is however not used in the Foroba Blon use case. While Adama is comfortable in the usage of a computer, he does not have any advanced technical skills, such as programming. However as he runs the radio station, he is familiar with processing audio fragments (using the open-source application Audacity).

The second iteration of the VSDK (which was used in this evaluation) included various bug-fixes, and added several features such as service elements that record user voice input, automatic configuration of Asterisk and audio file conversion. This evaluation has shown that it is possible for a local agent to develop and change elements in a voice service on the Kasadaka platform, achieving the goal of enabling locally owned and developed voice services.

During the session Adama has been instructed by the authors in the usage of the VSDK's development interface. Together we walked through the process of changing properties in the interface, adding new elements (such as new languages), recording and adding new voice fragments to the system and various other aspects. After this short training of about an hour, we asked Adama to go through the process again by himself, in order to verify that he was able to now use the VSDK on his own to change the properties of the voice service. This went successfully and Adama found the methodology and functionality of the VSDK to be well set up, and was satisfied with the way in which he was able to develop and maintain voice services through the development interface.

During the evaluation, Adama has successfully used the VSDK to apply and adapt the included voice service interaction templates to the Foroba Blon use case. Support for the Malian language Bambara (for which no TTS and ASR exists) was added to the system by recording voice fragments and adding them

through the development interface. The resulting service was tested using the local phone network and will be evaluated further during the programs of Radio Sikidolo. The radio will use the service in their interactive radio programming, in which the local population (which is has a low literacy rate) can participate by calling to the radio station (with their simple mobile phones) to ask questions or to state their opinions. The combination of the Kasadaka's hardware and the VSDK allow for the off line development and maintenance of voice services by Adama, who falls in the intended user group for the VSDK and the Kasadaka platform and thus does not have any programming knowledge or advanced computer skills (see Figure 4). During this case study the combination of hardware and software in the Kasadaka platform was successful in enabling the hosting and development of voice based information services in the context of the developing world. While it is still a case study and the outcomes are not guaranteed to be generalize-able, the outcomes show significant potential in bringing the advantages of the Internet to the world's disconnected populations. At the time of writing the Kasadaka platform is being refined iteratively and is regularly tested by Adama, who describes his experiences with the platform through phone calls with the authors.

Adama's level of computer literacy is around that of the targeted user group for the development of voice services on the Kasadaka platform. These voice service developers do not need to have programming skills, but some knowledge of using computers is required such as being able to use more complex web-based interfaces (such as a web-based e-mail client). In the future these users could then be trained (over several days) in the process of designing and developing voice services for local use cases.

6 RELATED WORK

This section covers existing efforts in the development of Web-extensions in the developing context, as well as tools and applications that facilitate the development of information services in low-resource environments.

Large-scale Voice Services. Voice based information systems that use the local (2G) mobile telephony network have already proven to be effective in reaching the rural population of the developing world. To support development of voice-based, mobile micro-

services Orange Labs developed the *Emerginov*[13] platform in 2012, targeting users in low resource environments such as e.g. rural Africa. It includes support for generation of voice-content in local languages, such as Wolof, a local language spoken in Senegal. Emerginov is normally hosted in the cloud, i.e. in a data center, connected to the Internet and the local phone network. Its hardware allows for 32 concurrent (in- or outbound) calls. Emerginov was technically promising, but the service has been discontinued by the operator after a successful pilot. (Gyan et al., 2013)

The company Viamo[14] runs several voice-based information services in many African countries. Viamo develops voice services for companies and NGOs. The company has contracts with several African telecommunication companies, allowing the local population to call these services without cost, using a toll-free number. The services Viamo develops are mainly aimed at large populations, with a very large number of concurrent calls. Although these services are able to reach a large amount of people, the large scale of the organization and the infrastructure that is required to run these large scale services, causes services services targeted at the rural poor to be financially unsustainable.

Twilio Studio[15] is a web-based application that allows graphical voice-service development by dragging and dropping interaction elements into a call flow, which are the components in a voice service. However the deployment of voice-services created in Twilio Studio is limited to the Twilio platform, which does not offer local phone numbers in many of the developing countries where voice-services could be relevant. This severely restricts the availability of the voice services on the Twilio platform. Twilio Studio seems to not be usable without an Internet connection, which can not be assumed to be available. Furthermore, just like the previous examples Twilio makes intensive usage of TTS and ASR technologies, which are not available in the languages spoken by the local population.

SMS-based Data Gathering Tools. In contexts where a connection to the Internet is not available, SMS can be used as a medium to exchange information with an automated system.

RapidSMS[16] is a tool set that allows for the development of SMS-based services for data collection and other work flows. RapidSMS is devel-

---

[13]See: https://emerginov.ow2.org/
[14]See https://viamo.io
[15]https://www.twilio.com/docs/api/studio
[16]https://www.rapidsms.org/

oped by UNICEF and has been used for various use cases, including remote health diagnostics and nutrition surveillance. RapidSMS is open-source and very scale able to suit large deployments, but can also run on a low-end server with a GSM modem. (Ngabo et al., 2012)

DataWinners[17] is a data collection platform that is developed by Human Network International[18] (HNI). DataWinners enables the development of SMS and smart phone based data surveys. These surveys are primarily aimed at the context of NGOs that need to retrieve data from their extension workers. By using SMS data can be collected without a need for an Internet connection, while the data can be still be entered through a user-friendly graphical interface on a smart phone. In the DataWinners web-based environment, new data surveys can be developed in a graphical interface.

Discussion. There exist several platforms for the development and hosting of large-scale voice services. These platforms allow for services that handle many concurrent calls and are thus well suited to services that aim to reach the general population. The drawback is that the infrastructure and development processes required for these services, are very expensive and thus out of reach of the local population.

While SMS-based services provide data exchange in contexts with limited Internet connectivity, it is only usable by the literate that have knowledge about the usage of SMS. Large populations in the developing world are illiterate or do not know how to use SMS. Thus while SMS-based services work well for data exchange without the Internet, these services are not accessible for the general population in the developing world.

In conclusion, the existing solutions for the hosting and development of voice services and SMS based information services are not capable of providing benefits that are comparable to those of the Internet, at a cost that allows for financially sustainable voice services in the developing context. Besides the issue of cost, other problems for the application of these solutions in the context described in this article lie in the area of support of under-resourced languages, the centralized nature of these solutions, and the requirement of a reliable connection to the Internet.

7 CONCLUSION

The wider aim of the presented Kasadaka platform and its Voice Service Development Kit is to allow the populations on the other side of the Digital Divide to share knowledge and create content, analogous to the advantages that the Web provides. The platform is lightweight and is tailored to the harsh circumstances that are found in the Global South and takes into account the information needs of the local population. By enabling local voice service development and making custom voice services affordable for the world's rural poor, Kasadaka enables the formation of a network of decentralized voice services. Such a network has the potential to provide the benefits of the Internet to the rural poor, reducing the gap of the Digital Divide and helping to improve the quality of life and well-being in the developing world.

Future work on the platform will focus on further expanding the voice service development functionality as well as more sophisticated data management, to allow for the development of more complex voice services. The VSDK is currently still too limited for applications that enable complex data exchange, and currently still requires writing code in order to support these more demanding use cases. Furthermore, the hardware of the platform is to be made more robust to better withstand the conditions in the developing context. Other ideas on further expansion include the implementation of a TTS system that is suitable for under-resourced languages, solving the dependency on the closed source VoiceXML browser and allowing for the inclusion of external data sources that are available on the Internet.

Although there is still a long way to go, we believe we have made plausible that for Web-like information and knowledge exchange, alternatives to simple technology transfer are possible that do much more justice to the local realities and needs at the other side of the Digital Divide.

17https://www.datawinners.com/
18http://hni.org/

REFERENCES

Ali, M. and Bailur, S. (2007). The challenge of sustainability in ICT4D-Is bricolage the answer. In *Proceedings*

of the 9th international conference on social implications of computers in developing countries. Citeseer.

Bagshaw, P., Barnard, E., and Rosec, O. (2011). VOICES Deliverable D3.1: Report on state of the art and development methodology. Technical report.

Berment, V. (2004). *Méthodes pour informatiser les langues et les groupes de langues «peu dotées»*. PhD thesis, Université Joseph-Fourier-Grenoble I.

Besacier, L., Barnard, E., Karpov, A., and Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56(Supplement C):85 – 100.

Black, A. W. and Lenzo, K. A. (2000). Limited domain synthesis. Technical report, Carnegie-Mellon University.

Bon, A., Akkermans, H., and Gordijn, J. (2016). Developing ict services in a low–resource development context. *Complex Systems Informatics and Modeling Quarterly*, (9):84–109.

de Boer, V., Gyan, N. B., Bon, A., Tuyp, W., Van Aart, C., and Akkermans, H. (2015). A dialogue with linked data: Voice-based access to market data in the sahel. *Semantic Web*, 6(1):23–33.

de Vries, N. J., Davel, M. H., Badenhorst, J., Basson, W. D., de Wet, F., Barnard, E., and de Waal, A. (2014). A smartphone-based asr data collection tool for under-resourced languages. *Speech Communication*, 56(Supplement C):119 – 131.

Farrugia, P.-J. (2005). Text to speech technologies for mobile telephony services. *Pace and Cordina [PC03]*.

Fuchs, C. and Horak, E. (2008). Africa and the digital divide. *Telematics and Informatics*, 25(2):99–116.

GSMA (2016). GSMA report: The Mobile Economy: Africa.

Gyan, N. B., de Boer, V., Bon, A., van Aart, C., Akkermans, H., Boyera, S., Froumentin, M., Grewal, A., and Allen, M. (2013). Voice-based web access in rural Africa. *Proceedings of the 5th Annual ACM Web Science Conference on - WebSci '13*, pages 122–131.

Heine, B. and Nurse, D. (2000). *African languages: An introduction*. Cambridge University Press.

Krasner, G. E., Pope, S. T., et al. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3):26–49.

McTear, M., Callejas, Z., and Griol, D. (2016). Creating a Conversational Interface Using Chatbot Technology. In *The Conversational Interface*, pages 125–159. Springer.

Ngabo, F., Nguimfack, J., Nwaigwe, F., Mugeni, C., Muhoza, D., Wilson, D. R., Kalach, J., Gakuba, R., Karema, C., and Binagwaho, A. (2012). Designing and implementing an innovative sms-based alert system (rapidsms-mch) to monitor pregnancy and reduce maternal and child deaths in rwanda. *The Pan African Medical Journal*, 13.

Poushter, J. (2016). Smartphone ownership and internet usage continues to climb in emerging economies. *Pew Research Center*, 22.

Rolland, C. (2007). Capturing system intentionality with maps. In *Conceptual modelling in Information Systems engineering*, pages 141–158. Springer.

Schmida, S., Bernard, J., Zakaras, T., Lovegrove, C., and Swingle, C. (2017). *Connecting the Next Four Billion: Strengthening the Global Response for Universal Internet Access*. USAID, Dial, SSG Advisors.

The World Bank Group (2016). Digital dividends, world bank development report. Technical report, The World Bank, Washington, US. DOI: 10.1596/978-1-4648-0728-2.

UNESCO (2011). UNESCO report: Regional overview: sub-Saharan Africa.

USAID (2017). Closing the access gap: Innovation to accelerate universal internet adoption. Technical report.