Affordable Voice Services to Bridge the Digital Divide – presenting the Kasadaka platform

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

[1] Amsterdam Business School, Universiteit van Amsterdam, Amsterdam, The Netherlands
a.h.j.baart@uva.nl
[2] the Network Institute, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
{a.bon, v.de.boer, w.tuijp, j.m.akkermans}@vu.nl
[3] Centre for International Cooperation, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
[4] University of Development Studies, Tamale, Ghana
fdittoh@uds.edu.gh

Keywords: digital divide, low literacy, sub-Saharan Africa, voice-based services, low-resource hardware, services development software kit

Abstract. Despite its global reach, the World Wide Web still fails to serve about 3 billion people, the majority living in the Global South, especially in poor, low-resource regions, where broadband internet is not expected to be rolled out in the foreseeable future. Yet, to bring the advantages of ICTs at the reach of communities in low-resource development regions, lightweight, affordable and context-aware ICT solutions are needed, that fit local needs and context. To this end the *Kasadaka* platform was developed. This platform supports easy creation of local-content and voice-based information services, targeting currently 'unconnected' populations, taking into account contextual and infrastructural requirements, and matching local ecosystems. The Kasadaka platform and its Voice Service Development Kit support the development of decentralized voice-based information services, to serve local populations and communities in their own local languages, 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 public open space for knowledge sharing, content creation and application service provisioning for billions on this planet. Despite its global reach, the Web is not yet accessible for three billion people in the world, the majority of whom live in the Global South, often in remote rural regions, under low-resource conditions and with poor or even absent infrastructures [11]. Obviously, the right to share knowledge and access relevant content should not be denied to people who live at the "other side of the Digital Divide".

Omnipresence of internet connectivity and large scale transfer of technologies from the Global North to poor regions, is generally assumed to be the best solution to this global problem. To this end large funds are made available by international development donors[5].

For example in Mali, one of the poorest countries in the world, 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 low levels of literacy in the population (around 35% on average, for women even lower).[6] Under these conditions it is unlikely that technology transfer for Internet roll-out will take place in the foreseeable future.

An alternative approach to large-scale technology-transfer is an approach focusing on development of light-weight community-centered ICT services, based on locally expressed needs and embedded in local ecosystems. In this research we focus on the co-creation of new, context-aware services for information exchange and knowledge sharing support targeting specifically farmer communities in the West African Sahel.

In this paper we show that services can be developed that enable access to information and knowledge exchange in low-resource contexts, in the absence of Internet connectivity. This requires thorough field investigation to assess conditions, requirements and specific conditions. Context analysis leads to new insights and gives technical directions, which cannot be derived from advanced but faraway technology considerations alone. This approach stands in contrast to the one-size-fits-all technology transfer approach that is common in international development projects [24,21,22].

In this paper we present *Kasadaka*, a platform that intends to support the hosting and development of locally relevant voice-based information services, targeting 'unconnected' populations and meeting the harsh conditions at the "other side" of the Digital Divide.

The Kasadaka platform and its Voice Service Development Kit aims to serve local communities and small businesses, by facilitating the formation of a local ecosystem of decentralized voice-based information services, analogous to the services of the Web, but without need for high speed infrastructure or Internet connectivity.

This paper is an extended version of the paper presented at the 14th International Conference on Web Information Systems and Technologies (WEBIST18) [3]. This publication extends the preceding paper by including three additional evaluations of the Kasadaka platform, a description of an additional feature of the VSDK, a revised introduction and a revised and extended section on methodology and approach.

This paper is structured as follows: Section 2 describes the methodology and approach used to elicit the requirements for designing and building the the platform. Section 3 present the resulting requirements from the requirements analysis, Section 4 describes the architecture and technical implementation, and Section 5 describes various evaluations of the platform. Sec. 6 discusses related work and Section 7 summarizes our main conclusions.

[5] https://webfoundation.org/our-work/projects/alliance-for-affordable-internet/
[6] http://uis.unesco.org/en/country/ML

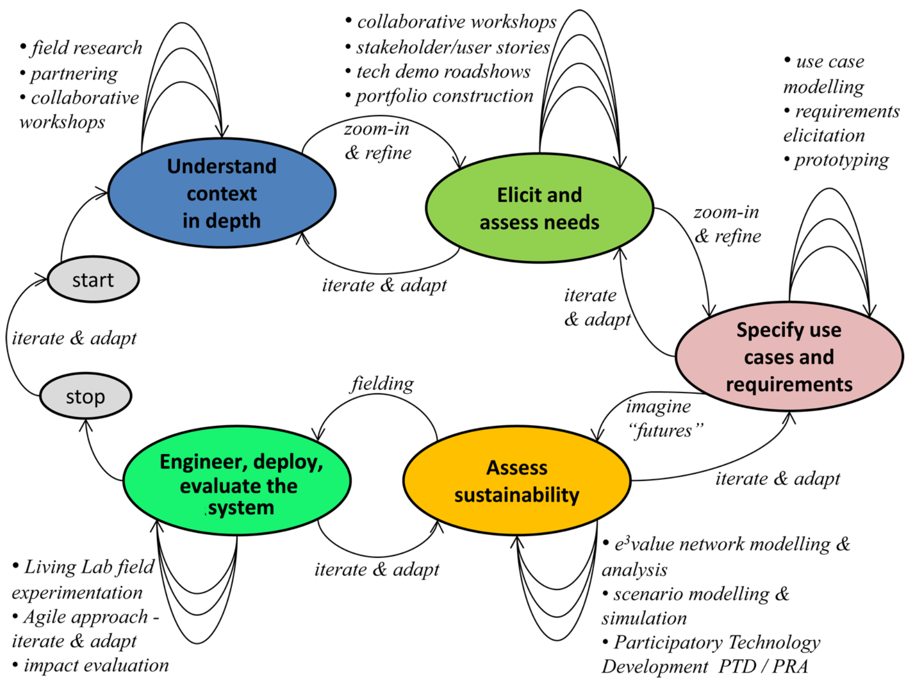

Fig. 1. ICT4D Field research methodology, from [9].

2 METHODOLOGY AND APPROACH

Whereas, in a high-tech environment in the Global North, setting up the requirements for a new ICT service platform can be considered a well documented business-as-usual task, in which technical specifications are balanced against costs and expected revenues, this is less straightforward for platform design that targets low resource environments such as e.g. rural Africa. While adaptation to local contextual conditions is to be expected, these conditions are often initially unknown to the platform designers. In this section we present an adaptive, iterative field-based approach and methodology, on how to elicit platform requirements for complex low resource contexts. This task must be performed in close collaboration with local envisaged users of the platform, to ensure matching local needs.

For the whole life cycle of Information Systems engineering in the context of ICT for Development (ICT4D), we use an iterative, adaptive and collaborative field research methodology, based on extensive field piloting. This is depicted in Figure 1, in the form of an intention-strategy map [20]. The methodology, dubbed ICT4D 3.0, has been extensively tested in low-tech low-resource environments, in the period 2012 - 2018 in rural regions of West Africa [9]. The methodology consists of various stages, not necessarily to be performed in a fixed order:

– An analysis of the local context: a through field-based analysis of local livelihoods, to find out local conditions with respect to technical infrastructure, environmental conditions, availability of technical support, etc;

– A collaborative needs assessment to find out the platform requirements, as also derived from the service use cases, elicited from local users during extensive user workshops;
– In-depth use case and requirements analysis in collaboration with local users, during workshops with focus group discussions and group assignments. This consists of elicitation of user and business requirements for the services, and technical, contextual and business requirements for the platform.
– Technical design, prototyping, engineering, piloting, deployment of the platform in the local environment; iterative testing and doing user evaluations; this may lead to new requirements and second cycle improvements.
– Analysis of the local ecosystem in which the platform is supposed to function; this is necessary to predict economic sustainability, which can be determined by analysis of the network of agents that exchange objects and services of value with each other, in order to deliver a service to the envisaged end users/customers. This analysis can be done in an early stage, before the actual implementation/deployment of the platform, and must involve data on the expected service (size and cost of the services, number of users, all expected costs involved).

3 PLATFORM REQUIREMENTS ANALYSIS

The development of a system that is intended for people at the "other side of the Digital Divide", has to deal with circumstances and issues that are rarely encountered in technology development projects in the developed world. Therefore, finding requirements for a service platform is done in a iterative approach in which analyzing the requirements for the use cases and the service platform requirements go hand in hand. Figure 2 shows the various sources where requirements have been derived in the case of the Kasadaka platform.

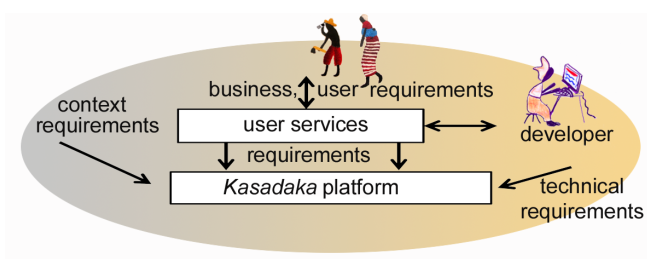

Fig. 2. Different sources for platform requirements: (i) contextual requirements related to e.g. cultural and language, (ii) technical requirements related to e.g. infrastructure, (ii) user and business requirements, derived from use case and requirements analysis that lead to the design and deployment of local user services.

3.1 Requirements related to the local context

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 puts the vast amounts of textual information on the Internet out of reach for a major part of the population in these regions [23]. Furthermore, many indigenous cultures have a strong oral tradition in communication, so that voice-based services have a natural fit with the locally 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 has around 2000 local languages, which each often have local dialects [14]. 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 assistants 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 research and a substantial financial investment in order to support a language at a level that is sufficient for usage in voice services. [4,7,10,17,6,25] The number of languages that has well-developed speech technologies is rising, but these do 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 [5].

3.2 Technical requirements

Information services for the world's poor should be affordable and accessible through locally adopted technologies, i.e., 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.[7] 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 by using a device they already own or have access to. The initial costs and running costs of a voice service should also be low enough to be affordable (and to provide sufficient return on

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

investment) for the rural poor.

The voice-service platform should function with limited infrastructure.
The (digital) infrastructure in these countries is often 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 (even in cities). The majority of the population does not have (direct) access to electricity[8]. The ownership and usage of smartphones that can access the Internet is slowly becoming more common [19], but internet connections are still very expensive and unreliable[9], 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, and has become the main means of telecommunication in sub-Saharan Africa [12]. The coverage of mobile telephony networks is often quite good, covering a large part of the population (also rural areas).

3.3 Business requirements, financial sustainability and the local ecosystem

The platform should be able to provide financially sustainable voice services.
This can be 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 accurately 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 in developing countries owns (or has access to) a computer, let alone a connection to the Internet. As a consequence, 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 (in the financial context of these countries), 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, as it keeps development costs at a minimum. 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 of the development process should

[8] `https://data.worldbank.org/indicator/EG.ELC.ACCS.ZS?end=2014&locations=ML&start=2014&view=map`

[9] As an illustration to this point, a group of Malian entrepreneurs launched a media initiative to persuade local Internet Service Providers to provide better services: `http://100mega.ml/`

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 of sourcing developers locally 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 further aids in ensuring the local relevancy of the services as well as the understanding of the end-user's needs. The platform should facilitate the founding 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 translates to "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 [1]. 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 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 be able to facilitate. These use cases have been elicited and analyzed during our various field visits to Mali, Burkina Faso and Ghana.

Foroba Blon, a system for village reporting.
We briefly describe here the case of Radio Sikidolo, a small radio station in Konobougou, a village in the south of Mali several hours from the 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 [13]. 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 5.5

Weather information
Many farmers and families in sub-Saharan Africa 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.

The request of weather information has been repeated in many of the communities visited by members of the W4RA team. A similar variant (although with a slightly different context) of the use-case was elicited in Guabuliga, a village in the Northern Region of Ghana. Farmers in the community, upon understanding the broad concept of voice-services; providing information in their own language, immediately suggested weather forecast information. In Ghana, accurate seasonal (regional) rainfall forecast is available from the National Geological Services mostly online and during Television Weather Reports (in English). Less accurate, but usable daily and weekly local forecasts are also available through a combination of satellite data and local weather stations which feed open weather sources online. The lack of access to these sources of information, due to the lack of internet and low literacy (inability to read and write in English) results in this information not reaching members of these communities despite the fact that it remains relevant to them, for similar reasons as covered in the example of Burkina Faso. The Ghanian version of the weather information voice-service –dubbed *Mr. Meteo*– makes use of open-weather data APIs to source it's weather data.
The Mr. Meteo service was built with the VSDK and was used in an evaluation of the Kasadaka platform, see Section 5.4.

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 hard-

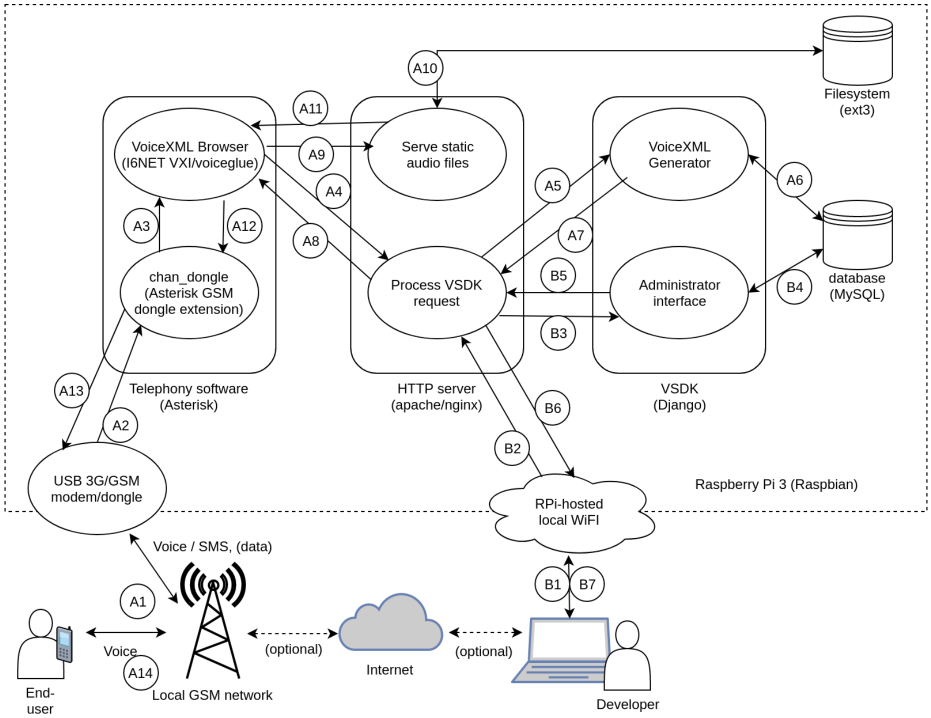

Fig. 3. Overview of the Kasadaka system architecture.

ware and accompanying software. Figure 3 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[10] (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 (which can also connect to 2G networks) is used. The exact

[10] A Raspberry Pi 3 (including case, power supply and SD card) costs around €60 at the time of writing.

make and model of this modem can differ, as long as it is on the supported hardware list[11] of the `chan_dongle` Asterisk extension.

4.2 Software

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

Telephone exchange software: Asterisk
Asterisk is a very popular open-source Private Branch Exchange (PBX) telephony application. It is used for the routing of incoming calls to its destination using Voice-over-IP technologies. In the implementation 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`[12], 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 and USSD codes.

Voice application document standard: VoiceXML
VoiceXML[13] 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 spoken sentences, 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[14], a closed-source VoiceXML interpreter built by the company I6NET[15]. 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

[11] `https://github.com/bg111/asterisk-chan-dongle/wiki/Requirements-and-Limitations`
[12] `https://github.com/bg111/asterisk-chan-dongle`
[13] `https://www.w3.org/TR/voicexml21/`
[14] `http://www.i6net.com/technology/voicexml-ivr/`
[15] `http://www.i6net.com/`

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 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[16] was chosen as the Python-based web-framework, as it has very good and extensive documentation, is well-supported and follows a Model-View-Controller (MVC) methodology [16]. 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 Voice Service Development Kit is the main software component in the Kasadaka platform. The main goal of the VSDK[17] 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 the requirement of 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 then interprets this VoiceXML file and 'displays' it to the user. In Figure 3 a visual representation of the data flows in the Kasadaka platform is shown.
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

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

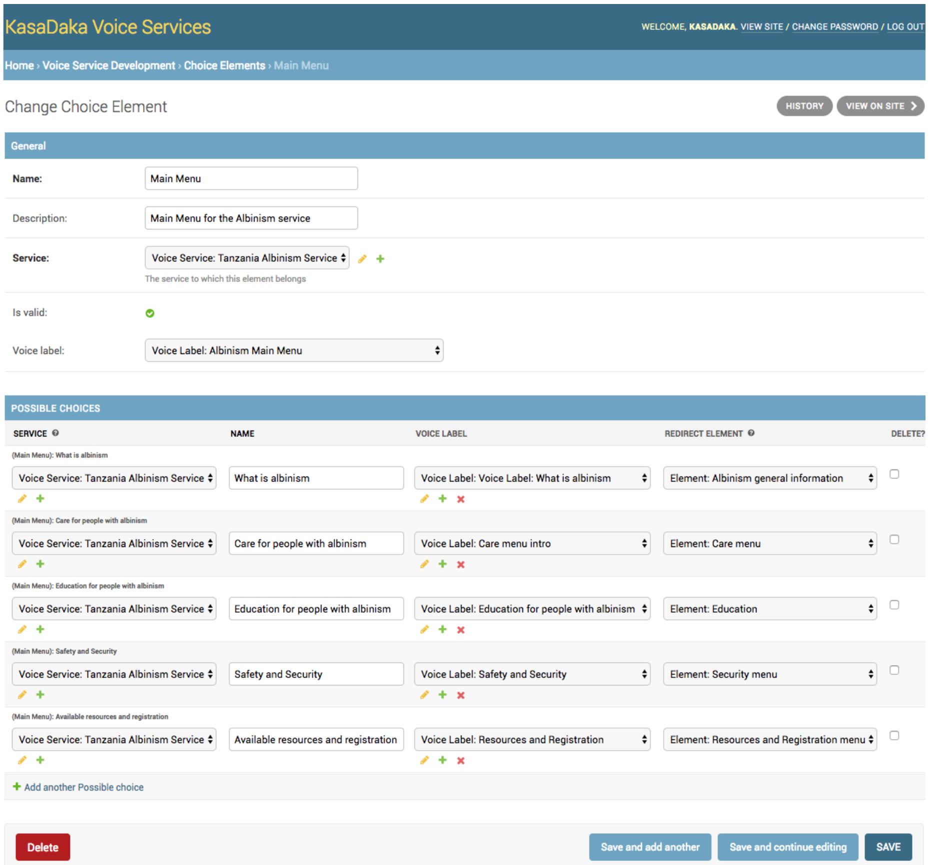

Fig. 4. An example screen shot of the voice-service development interface of the VSDK. Shown is an example of a choice interaction element. A voice service developer uses this GUI to develop voice-based applications on the Kasadaka platform.

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[8].

4.4 Bip recognition

In order to support certain use-cases, it is required that the Kasadaka system is able to recognize so-called *bips*, which can be compared to missed calls. In order to support the use-case as described in Section 5.2, the authors implemented the recognition of missed calls into the Kasadaka. The implementation consists of some elements in the Asterisk configuration which –instead of immediately accepting an incoming call– add some delay before accepting an incoming call. The time of the initial "first ring" is recorded in a logging file. Then the line will keep ringing for approximately 5 seconds, after which the call is accepted and the user will hear the configured voice-service. When the call is "picked up" by the system, another entry in the log file is made. By periodically parsing the log file, and recognizing the instances where a call was received, but was cancelled before the system established the connection, bips (missed calls) can be recognized by the Kasadaka. The periodical job that recognizes the bips can be configured to request a customize-able URL when a new bip has been recognized. This URL can be set to an endpoint in a VSDK application, which can then further process the bip according to a specific use-case.

5 EVALUATIONS

There have been several evaluations of the VSDK, with varying degrees of sophistication. The first evaluation was done in the Netherlands with inexperienced users of the VSDK, which was used to further refine the VSDK. Subsequent evaluations enriched our understanding of the limitations of the current state of the VSDK, and how it could be extended to support more complex and more varied use cases.

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. [2] The groups each developed a voice service for one of several use cases, which were co-created with rural communities and relevant in the context of the developing world and provided to the students in written form. The choice to evaluate with students was made because of the ease of communication with the students, which is significantly less complex and expensive than traveling 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 underlying concepts and ideas of the VSDK's development work flow. The VSDK proved to be sufficient in providing the required functionality for the creation of basic voice-service services, and 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 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 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 area 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 lack of support for 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 to extend the included features of the kit. 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. 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 the inner workings, maintaining the advantage of ease of use offered by the VSDK.

5.2 Case study: BipVote, a voting system for rural Mali

During the successive edition of the ICT4D course at the Vrije Universiteit Amsterdam (see Section 5.1) in 2018, students received additional guidance on using the Django framework, prior to learning to develop voice services using the VSDK. This adjusted approach resulted in many student groups developing more complex applications that had both voice and web-based interfaces, extending significantly on the basic included functionalities of the VSDK. We will cover one of the applications here in depth, which showcases the creative uses and the diverse potential of voice services.
We present the case of the BipVote application[15], which was designed and built by ICT4D students at the VU University Amsterdam: Hans-Dieter Hiep, Roy Overbeek and Paweł Ulita. BipVote is a system that allows for polls to be held, in

which votes can be cast by sending so-called *bips*. A bip a word used in several French speaking west-African countries, to describe what could be considered a missed call. More specifically, a bip is a call that is cancelled quickly, before the recipient has had the chance to pick up. This is registered as a missed call on the recipient's phone. Because there has not been a phone connection, sending a bip is free of charge. Sending bips is common practice when one does not have enough money for a call, or when someone wants to send a signal; For instance, it can be a request to the recipient of the bip, to call the sender back.

During our visits and collaborations with local radio presenters, an additional usage of bips emerged, being that radio stations use bips in interactive radio programming, allowing listeners to cast votes without cost. These polls are held to gauge the public's opinion on social issues, but also as small quizzes that are aimed to estimate whether (in the case of educational programming) listeners have gained a sufficient understanding of the subject matter.

The way radio organizers organize the votes, is that they broadcast 2 phone numbers –one for voting yes, one for no– that listeners can send bips to. [18] Because programming is usually recorded and broadcast several times during a week –allowing listeners to tune in when convenient– the votes usually run for a week's time. At the end of the week –during the next episode of the program– the number of missed calls on the phones are manually counted and the results broadcast and used in future programming, for example to explain a subject in further detail. In the case of Radio Sikidolo (see Section 5.5) during a one-week poll, usually around 800 votes are cast.

BipVote streamlines this work-flow by providing both a web-interface and a voice-interface[19] to allow for managing polls and viewing (past) results. Radio presenters are able to set up new polls, entering the time-frame in which the poll will run. During and after the vote, the votes are counted and plotted on a graph. The presenters are able to see the results of polls develop in real-time. These same features are available through a voice interface, to allow management of polls when a computer is not available.

In order to realize the features of the BipVote system, the students significantly extended the VSDK with additional functionality to support the handling of bips (using the bip recognition feature outlined in Section 4.4), the management of polls through the web interface, as well as through the voice interface. The included *building-blocks* of the VSDK were extended with elements that tell how much time is remaining in a poll, that allow for setting the time duration of a new poll, creating new polls and removing polls. The element that informs the user of the results of a poll, includes a custom-built algorithm that allows for the programmatic conversion of numbers into their spoken representations, in multiple languages (English, French and Bambara are supported). This is achieved by combining samples to compose larger numbers. "For example 152 becomes *one-hundred-fifty-two* (4 samples) in English, and *cent-cinquante-deux* (3 samples) in French, and *keme-ni-bi-duuru-ni-fila* (6 samples) in Bambara." [15]

The BipVote use-case is an example of a service that enables creative use of older technologies (*bricolage*, see Section 3.3) that is prevalent across developing countries. One could imagine several other possibilities that bips could offer in

[18] For this use-case a Kasadaka can be equipped with two dongles, providing two telephone lines.

[19] A demonstration of the BipVote application is available on YouTube: `https://www.youtube.com/watch?v=mxtRCMht0qg`

other use-cases, such as: letting the Kasadaka call back an user (shifting the costs of calling to the owner of the Kasadaka), or low-cost signalling applications.

5.3 Case Study: Tanzania Albinism Society

The VSDK's suitability as a rapid prototyping tool has additionally been evaluated in Tanzania with the cooperation of Tjitske de Groot (Vrije Universiteit Brusssels) and Gamariel Mboya (Tanzania Albinism Society). Tjistke approached the authors with the wish of a voice-based system that could be used to educate people about albinism and collect contact information of people with albinism. Tjitske's research is centered around designing effective stigma reduction interventions for albinism in Tanzania. Gamariel and Tjitske had been looking into the possibility of running an experiment with a voice service that could educate the general public about albinism, and provide an alert system for those with albinism. Because of the costs associated with setting up a voice service at established companies –for which they did not have sufficient funding– they asked the authors whether it would be possible to build a low-cost prototype of the service. The authors were sent a short overview of the intended functionalities of the voice service, which was processed into a tree that describes the structure of the voice service. From this tree a list of voice fragments was compiled, which were subsequently translated to Swahili by Gamariel. As Tjitske and the authors were both in the Netherlands at the time of developing the prototype, it was not possible to set up a session with Gamariel to record the voice fragments for the service, as the authors often do during the process of voice service development. This was resolved by resorting to Gamariel sending us recordings of each of the voice fragments through the smartphone messaging app Whatsapp. These recordings were transferred and corrected for volume and background noise using Audacity, an open-source audio editing application. After the quality of the recordings was sufficient, the voice service prototype was developed using the VSDK web-interface, based on the structure of the voice-service tree. The authors showed Tjitske around the interface of the VSDK, explained the properties of the *building-blocks* available in the interface and created the main elements of the voice-service structure. Using the explanation and examples, Tjitske was able to learn developing voice-services using the VSDK by doing; By finishing the voice-service prototype herself.

Tjitske acquired a Raspberry Pi and dongle, on which the Kasadaka software stack was installed. The authors explained how to set up the system, and how to troubleshoot in case of problems. Tjitske took the Kasadaka to Tanzania, and was able to demonstrate the voice service to Gamariel without any issues.

In this evaluation the Kasadaka platform was used to develop and demonstrate a voice-service prototype with a low investment of time and capital. The nature of the intended voice-service made the Kasadaka unsuitable for a roll out. This is because the service targets the whole of Tanzania and thus should be able to support a high number of concurrent calls, which is not achievable on the Kasadaka, as it runs on low-resource hardware which supports only one concurrent call. The development cooperation with Tjitske has shown the potential for voice-service development training using the VSDK, and that the recording of voice fragments does not require a native speaker to be physically present, but can also be done remotely through sending voice messages.

Fig. 5. Gamariel Mboya calling the Tanzania albinism voice-service prototype, running from the Kasadaka on the table.

5.4 Case study: Mr. Meteo

The Mr. Meteo voice-service (see Section 3.4), provides community-specific weather forecast/information via voice prompts in local Ghanian languages. The system was deployed using the Kasadaka platform in a test phase in December 2018 at Bolgatanga in the Upper East Region of Ghana, to a cross-section of animal and crop farmers, gathered from 4 local communities. The deployment was in collaboration with Cowtribe[20], a company that provides veterinary services with the use of ICT-driven technologies, and was received with overwhelming enthusiasm.
The Mr. Meteo use case in Ghana, provides an evaluation of the Kasadaka platform in a real developing world context. The system was built using the VSDK, at the University for Development Studies in Tamale, Ghana by a team with programming backgrounds who had previously been involved in the development and student evaluation process of the VSDK. The validation session was done within the local community, with no internet access and using a power-bank as source of power. This is because Cowtribe –whose clientele tested the system– require a model where it is possible to host devices within the communities; hosted by trained community members. Development of the service however was done with the availability of electricity and internet (although internet access was not directly required). The audio fragments were recorded locally (in Gurune, the major language in Bolgatanga, for which no TTS and ASR exists) using a normal Android-based smartphone in a quiet environment and sent via internet to Tamale where it was converted to the appropriate formats and integrated into the VSDK. An additional language (Dagbani, also without available TTS and ASR) was recorded in Tamale, also using a smartphone and integrated into the system to provide a control test case in terms of language. The VSDK prompts were set

[20] https://www.cowtribe.com/

up to create a voice system that would welcome the user to the service and then proceed to the weather forecast for the current day as well as the subsequent day.

A total of roughly 5 hours was needed to completely develop the application. One minor challenge faced was the need for interfacing external data (weather forecasts in this case) which must be accessed daily and parsed to indicate the appropriate voice fragments to play to the user. For the test phase this was not necessary, but a future implementation will require a minor extension to the VSDK to include this functionality. The resulting system was tested using a local Ghanaian Network Provider (MTN) and further piloted to farmers from 4 communities in the Upper East Region of Ghana. Farmers were of the opinion that this was very useful and would be something they would use (some regularly, others occasionally) and also inquired of the availability of other types of information such as; disease outbreaks, human and animal health, farming practices and information on their children's schooling. The system is currently being further developed to cater for these 4 communities. This will therefore require a prompt to select the appropriate community, or will require pre-registration of numbers with assigned communities. This evaluation shows the potential of the rapid prototyping and deployment of locally relevant voice-services using the Kasadaka platform. While the decentralized nature of the Kasadaka platform offers advantages in communities' ownership of the services, it also poses challenges, such as the lack of availability of an internet connection, if the service requires frequent information retrieval from the internet.

Fig. 6. Francis Dittoh, the Cowtribe team and the farmers that participated in the testing of the Mr. Meteo voice-service in Ghana.

5.5 Case study: Radio Sikidolo

The Foroba Blon use-case was built using the VSDK, and 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 other evaluations were sufficient for a general validation of the methodology of the VSDK, they did not evaluate the VSDK in the intended context of a developing country, with a user that had limited experience in voice-service development. This evaluation addresses these limitations: it evaluates the VSDK and the Kasadaka hardware and software as a whole, in the intended developing world context, with users that match the intended user profile. 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 relatively 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).

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. 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. The VSDK was a significant improvement over the prototypes that the authors had previously tested in cooperation with Adama.

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 future visits of the authors to Radio Sikidolo. 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 7). 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 allowing non-software developers to maintain voice-services that are hosted at a remote location.

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. The design and set-up of such training sessions is to be determined in future work.

5.6 Discussion of results

The general results from the evaluations of the VSDK and the Kasadaka as a whole show potential. The VSDK has been proven to successfully allow the maintenance and development of simple voice-services in the ICT4D context. Developing voice-service prototypes can be done in a matter of hours, which is invaluable during workshops and requirements analyses in the field.
The Kasadaka platform is affordable and is suitable for deployment in remote locations. While Kasadaka is also able to host more complex applications, the development process of such applications still requires programming skills and more time. In order to be able to realize the goal of facilitating local entrepreneurship in the development of voice-services (without software developers), these limitations will have to be overcome.
While the current version of the VSDK proves the potential of web-based voice-service development, this functionality is still limited to simple use-cases only. More complex use-cases can be supported by the VSDK's foundations, but still require software development skills to extend the VSDK with features that are specific to the use-case. 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. 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.

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*[21] 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.

[21] See: `https://emerginov.ow2.org/`

Fig. 7. Adama Tessougué of Radio Sikidolo shows the Kasadaka on which the Foroba Blon voice service now runs with its Bambara language interface.

Fig. 8. André Baart and Adama Tessougué evaluating the VSDK running on the Kasadaka platform, at Radio Sikidolo in Mali.

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. [13]

The company Viamo[22] 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[23] 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[24] is a tool set that allows for the development of SMS-based services for data collection and other work flows. RapidSMS is developed 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. [18]

DataWinners[25] is a data collection platform that is developed by Human Network International[26] (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

[22] See `https://viamo.io`

[23] `https://www.twilio.com/docs/api/studio`

[24] `https://www.rapidsms.org/`

[25] `https://www.datawinners.com/`

[26] `http://hni.org/`

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.
Still, 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 are in the area of support of under-resourced languages, the centralized nature of these solutions and the requirement for 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 people at the "other side of the Digital Divide" to share knowledge and create content, analogous to the advantages provided by the Web. It takes into account the information needs of the local population, by enabling the hosting and development of voice services that cater to local use-cases. The Kasadaka platform is lightweight, tailored to the harsh circumstances that are found in the Global South, and enables the formation of a network of decentralized voice services. Such a network has the potential to provide the benefits of the Web to the world's rural poor.
Despite the moderate size and outreach of our research project, the Kasadaka platform evaluations have shown, as an alternative to high speed internet solutions and technology transfers, the potential of easy-to-learn, lightweight, affordable and context-aware ICT solutions that do more right to complex realities, context and needs of people "at the other side of the Digital Divide". The Kasadaka is targeted at the low-resource context as found in many developing countries. Kasadaka's main software component, the Voice Service Development Kit, enables to-be voice-service administrators to develop and maintain voice-services in the field. It overcomes the lack of skilled software developers by reducing the skill-set required for voice-service development. Evaluations of the VSDK have shown it's potential in providing the foundation for the development of creative voice-services, catered to the needs of communities in sub-Saharan Africa. Voice-service prototypes can be created with little effort, which enables rapid prototyping and demonstrations in the field.

ACKNOWLEDGEMENTS

The authors thank Christophe Guéret for the original idea and his initiative in the first steps towards the Kasadaka platform. Additionally, the authors would like to thank Adama Tessougué, Amadou Tangara, Gossa Lô, Julien Ouedraogo, Matthieu Ouedraogo, Hans-Dieter Hiep, Roy Overbeek, Paweł Ulita, Tjitske de Groot and Gamariel Mboya for their invaluable contributions to this research.

References

1. Ali, M., Bailur, S.: 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 (2007)
2. Baart, A.: Master Thesis - KasaDaka: a Sustainable Voice Services Platform. Tech. rep. (2017), https://www.kasadaka.com/img/master-thesis-andre-baart.pdf
3. Baart, A., Bon, A., de Boer, V., Tuijp, W., Akkermans, H.: Ney yibeogo - hello world: A voice service development platform to bridge the web's digital divide. In: Proceedings of the 14th International Conference on Web Information Systems and Technologies - Volume 1: WEBIST,. pp. 23–34. INSTICC, SciTePress (2018). https://doi.org/10.5220/0006893600230034
4. Bagshaw, P., Barnard, E., Rosec, O.: VOICES Deliverable D3.1: Report on state of the art and development methodology. Tech. rep. (09 2011)
5. Berment, V.: Méthodes pour informatiser les langues et les groupes de langues «peu dotées». Ph.D. thesis, Université Joseph-Fourier-Grenoble I (2004)
6. Besacier, L., Barnard, E., Karpov, A., Schultz, T.: Automatic speech recognition for under-resourced languages: A survey. Speech Communication **56**(Supplement C), 85 – 100 (2014). https://doi.org/https://doi.org/10.1016/j.specom.2013.07.008, http://www.sciencedirect.com/science/article/pii/S0167639313000988
7. Black, A.W., Lenzo, K.A.: Limited domain synthesis. Tech. rep., Carnegie-Mellon University (2000)
8. de Boer, V., Gyan, N.B., Bon, A., Tuyp, W., Van Aart, C., Akkermans, H.: A dialogue with linked data: Voice-based access to market data in the sahel. Semantic Web **6**(1), 23–33 (2015), http://www.few.vu.nl/{~}vbr240/publications/swj2013{_}deboer{_}dialogue.pdf
9. Bon, A., Akkermans, H., Gordijn, J.: Developing ict services in a low–resource development context. Complex Systems Informatics and Modeling Quarterly (9), 84–109 (2016)
10. Farrugia, P.J.: Text to speech technologies for mobile telephony services. Pace and Cordina [PC03] (2005)
11. Fuchs, C., Horak, E.: Africa and the digital divide. Telematics and Informatics **25**(2), 99–116 (2008). https://doi.org/10.1016/j.tele.2006.06.004
12. GSMA: GSMA report: The Mobile Economy: Africa (2016), https://www.gsmaintelligence.com/research/?file=3bc21ea879a5b217b64d62fa24c55bdf&download
13. Gyan, N.B., de Boer, V., Bon, A., van Aart, C., Akkermans, H., Boyera, S., Froumentin, M., Grewal, A., Allen, M.: Voice-based web access in rural Africa. Proceedings of the 5th Annual ACM Web Science Conference on - WebSci '13 pp. 122–131 (2013). https://doi.org/10.1145/2464464.2464496, http://0-dl.acm.org.oasis.unisa.ac.za/citation.cfm?id=2464464.2464496
14. Heine, B., Nurse, D.: African languages: An introduction. Cambridge University Press (2000)
15. Hiep, H.D., Overbeek, R., Ulita, P.: BipVote: A radio-based mobile voting system for rural Mali. Tech. rep. (06 2018), https://w4ra.org/wp-content/uploads/2018/08/Bipvote.pdf

16. Krasner, G.E., Pope, S.T., et al.: A description of the model-view-controller user interface paradigm in the smalltalk-80 system. Journal of object oriented programming **1**(3), 26–49 (1988)
17. McTear, M., Callejas, Z., Griol, D.: Creating a Conversational Interface Using Chatbot Technology. In: The Conversational Interface, pp. 125–159. Springer (2016)
18. Ngabo, F., Nguimfack, J., Nwaigwe, F., Mugeni, C., Muhoza, D., Wilson, D.R., Kalach, J., Gakuba, R., Karema, C., Binagwaho, A.: 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** (2012)
19. Poushter, J.: Smartphone ownership and internet usage continues to climb in emerging economies. Pew Research Center **22** (2016)
20. Rolland, C.: Capturing system intentionality with maps. In: Conceptual modelling in Information Systems engineering, pp. 141–158. Springer (2007)
21. Schmida, S., Bernard, J., Zakaras, T., Lovegrove, C., Swingle, C.: Connecting the Next Four Billion: Strengthening the Global Response for Universal Internet Access. USAID, Dial, SSG Advisors (2017)
22. The World Bank Group: Digital dividends, world bank development report. Tech. rep., The World Bank, Washington, US (2016), dOI: 10.1596/978-1-4648-0728-2
23. UNESCO: UNESCO report: Regional overview: sub-Saharan Africa (2011), `http://en.unesco.org/gem-report/sites/gem-report/files/191393e.pdf`
24. USAID: Closing the access gap: Innovation to accelerate universal internet adoption. Tech. rep. (2017)
25. de Vries, N.J., Davel, M.H., Badenhorst, J., Basson, W.D., de Wet, F., Barnard, E., de Waal, A.: A smartphone-based asr data collection tool for under-resourced languages. Speech Communication **56**(Supplement C), 119 – 131 (2014). https://doi.org/https://doi.org/10.1016/j.specom.2013.07.001, `http://www.sciencedirect.com/science/article/pii/S0167639313000915`