A Design Vocabulary for Scaffolding Group Interaction Archetypes through Synchronous Telephony

DAN RICHARDSON, Monash University, Australia
MD ADNANUL ISLAM, Monash University, Australia
BRONWYN J. CUMBO, Monash University, Australia
PRANITA SHRESTHA, Monash University, Australia
DELVIN VARGHESE, Monash University, Australia
TOM BARTINDALE, Monash University, Australia
PATRICK OLIVIER, Monash University, Australia

Multiple HCI projects have demonstrated the potential of digitally-enhanced, synchronous telephony platforms for use with and by resource-limited communities. However, these platforms were each designed to only facilitate a single archetype of community engagement, limiting their capacity for adaptation when contextual or stakeholder requirements change. This paper builds upon these projects to introduce a design vocabulary, grounded in a formal ontology describing the core components necessary to run adaptable, structured engagements through synchronous group telephony. Through a series of scenarios, we demonstrate how this design vocabulary can be used to: help design and communicate different models of synchronous audio engagements, describe existing technologies, and highlight other novel ways in which such platforms could be used. We discuss how while under-explored to this point, synchronous telephony platforms can be designed to orchestrate stakeholder engagements with a degree of flexibility previously impossible in remote, offline contexts.

CCS Concepts: • **Human-centered computing → HCI design and evaluation methods**; **User centered design**.

Additional Key Words and Phrases: IVR, design, resource-limited settings, ICT4D

1 INTRODUCTION

The value of technologies which support synchronous remote group communication has been reinforced in recent years, with the increased adoption of remote work, recreation, and education [8, 31]. However, those technologies are often inaccessible to communities who lack reliable access to digital infrastructure or high levels of literacy. Recent HCI research has explored the use of telephony to support resource-limited communities accessing the types of services typically associated with Internet-based infrastructures [7, 9, 19, 26]. While the majority of these Interactive Voice Response (IVR) platforms enable asynchronous communication (e.g. voice forums for discussion and knowledge sharing [24]), a recent focus has emerged on the use of telephony to enable synchronous group communication. These technologies have typically been used to infrastructure 'broadcast radio' style talk shows with underprivileged communities: supporting listeners with platforms for discussion and access to expert knowledge through audio interfaces, without requiring specialised equipment or Internet access [16, 35, 42].

However, while recent research has investigated the repurposing and reconfiguration of Internet-based technologies such as Zoom to adapt to diverse new use cases [2], the use of these synchronous telephony platforms beyond these 'talk show' formats has been limited: as they are described, each

Authors' addresses: Dan Richardson, dan.richardson@monash.edu, Monash University, Melbourne, Australia; Md Adnanul Islam, adnan.islam@monash.edu, Monash University, Melbourne, Australia; Bronwyn J. Cumbo, bronwyn.cumbo@monash.edu, Monash University, Melbourne, Australia; Pranita Shrestha, pranita.shrestha@monash.edu, Monash University, Melbourne, Australia; Delvin Varghese, delvin.varghese@monash.edu, Monash University, Melbourne, Australia; Tom Bartindale, tom.bartindale@monash.edu, Monash University, Melbourne, Australia; Patrick Olivier, patrick.olivier@monash.edu, Monash University, Melbourne, Australia.

appear to have been built to specifically accommodate a specific engagement structure, without consideration to adaptability or how emergent formats of audio interaction could be supported. The focus on these 'single use case' platforms, each designed to facilitate the same interaction archetypes, has meant that communities lacking access to more flexible online synchronous communication platforms such as Zoom are currently excluded from engaging in these emerging forms of remote participation. Furthermore, during the development of our own synchronous telephony platform, we realised that many of these systems are built upon similar technologies and underlying design approaches: we found we were combating a steep learning curve when interfacing with (frequently poorly documented) telephony software stacks, simply to re-tread ground already contributed in prior research projects.

As seen in other emerging HCI design contexts (such as human-IoT system interaction [4], or user identification through touch [17]), a lack of agreed upon context-independent terminology to describe components, behaviours and characteristics can inhibit designers engaging with the space—especially those who are non-technical [17]. Similarly, believe that there is a non-trivial barrier to entry for HCI designers and developers entering the space of synchronous telephony: that research has been limited by a lack of a cohesive vocabulary which differentiates between potential design spaces and the underlying technical architecture required to support them. Without this differentiation, subsequent projects have had to 'reinvent the telephony wheel' as we did: spending significant development time designing and building (very similar, nonnovel) underlying infrastructures, rather than concentrating on exploring the multiple design spaces these platforms could provide.

To address this issue, this paper contributes a design vocabulary for synchronous group telephony: a visual framework which, by utilising a concrete set of terms and relationships defined in a formal ontology, is designed to help prototype, communicate and integrate synchronous telephony designs. To illustrate how this vocabulary can assist in designing synchronous telephony platforms, we first demonstrate its application to the established talk show format used in prior research projects. Then, by applying it to other possible use cases, we discuss how such platforms can be designed to orchestrate remote stakeholder engagements with a degree of flexibility previously impossible in contexts with offline participants. We hope that by introducing a cohesive set of axioms and differentiation between interaction design and technical infrastructure, this design vocabulary will support future researchers in identifying and navigating the multiple rich, under-explored design spaces of synchronous group telephony.

2 RELATED WORK

2.1 IVR & Asynchronous Audio

By allowing users to interact with digital systems through standard phone calls using their phone's keypad (and sometimes even their voice [19]), Interactive Voice Response (IVR) systems have been shown to be an effective tool for information retrieval and exchange: especially in low-resource contexts where users have limited internet access [9] and low literacy inhibits the usability of text-based interfaces [32]. IVR and asynchronous audio has proven to be a versatile medium, with previous projects utilising it in a wide variety of contexts and use cases, including: performing voice-based surveys [18]; supporting users in accessing information on-demand [25]; connecting offline workers to online marketplaces [26]; enabling citizen journalism [10]; creating 'voice forums', where users can create and respond to audio messages through phone calls [24, 39]; and interacting with automated conversational agents ('chatbots') to access information [15, 41]. These systems offer greater accessibility to users from communities with oral traditions [22], and can be designed to support cost-free, on-demand access to end-users [26]. Additionally, prior studies have highlighted

the value of being able to access recordings for later reference on-demand [35], with the ability to repeatedly access the same information being useful for low-literate users who cannot write information down [15].

Despite the versatility and scalability afforded by this infrastructure, however, automated IVR systems aren't always able to fulfill users' highly specific and contextual queries: traditional IVR menu structures are limited in scope, requiring designers to choose what information and interactions should be prioritised [26]. Chatbots have been shown to be able to respond accurately to highly specific queries [15], but such systems are dependant on currently unreliable speech-to-text systems and access to extensive databanks of previous responses by human experts. Furthermore, for users unfamiliar with the limitations of chatbots, interacting with them can clash with expectations of interacting with a human over the phone [15]: resulting in conversational queries which humans could interpret correctly, but automated systems struggle with [41]. Meanwhile, voice forums offer human replies, but the ability for anyone to respond to queries can lead to misinformation [24], as well as harassment, threats, and systemic marginalisation towards women and minorities [40].

2.2 Radio & Synchronous Audio Platforms

Synchronous voice formats, such as radio, can address these issues through direct moderation of a synchronous activity by a host. Often run by volunteers, community radio stations broadcast live audio content to local communities: offering a method for civic engagement, deliberation and knowledge sharing [5, 20]. Projects such as RootIO have drastically lowered the costs of radio infrastructure and encourage community-led participatory programming [7], particularly when combined with technologies for generating audio content, such as text-to-speech [30]. Unfortunately, while many radio programmes offer the ability for audiences to call into a show to ask questions, by default radio is an inherently passive medium in terms of audience interaction. Furthermore, broadcast radio often requires the navigation of local policy and broadcast content standards: erecting legal, logistical and psychological barriers due to formal license applications and requirements [3, 20].

In recent years, researchers such as Kazakos et al. [16] have begun sidestepping these barriers by providing access to 'radio' programming through phone-based platforms: hosting live shows through synchronous telephony, with listeners able to dial in with their phones to listen and ask questions to a host controlling the show from a smartphone or web application. These platforms use free, 'off the shelf' software such as FreeSWITCH[1] to programmatically manipulate phone calls. To support structured discussion, listeners join the call muted, and can request to be unmuted by the host by pressing a button on their handset (sending a 'DTMF' signal tone to the system, highlighting the caller on the host's display). Yadav et al. note the potential for such platforms to go beyond the 'radio' model by additionally supporting asynchronous pre- and post- show interactions, where the audience could dial into the system to submit questions or relisten to prior content on demand [42]. As Kazakos et al. (and follow-up investigations by researchers using similar systems [35, 42]) demonstrate, synchronous telephony allows for remote group engagements with a low barrier to entry, and offers the potential for active audience participation: bringing education, training, and individuals' specific expertise into the homes of rural and underprivileged communities, for whom such knowledge might otherwise have been inaccessible.

Despite this promise, however, these platforms still feature significant design limitations: taken as described in their publications, they are all configured exclusively around the 'talk show' concept, with the call host primarily leading discussions with a domain expert and fielding questions from the (otherwise muted) listening audience. These projects have highlighted a number of challenges

[1] https://signalwire.com/freeswitch

with this format: for example, a dependence on questions can lead to 'content black holes' when they are not received [16]; and while this can be mitigated through pre-prepared audio content [42], an over-reliance on recorded audio and muted participants doesn't take full advantage of the dynamic possibilities of synchronous audio, nor the two-way nature of typical telephone interactions. Furthermore, there is an asymmetrical power imbalance embedded in the talk show design pattern: the call's host has a visual overview and control over the call and its participants, while others—outside of requesting to be unmuted—are relegated to being passive consumers [38]. This would make the format unsuitable in contexts which value more open, free-flowing and unbiased discussion [11], or those that require structured procedure to allow for equal opportunities for participation [27]. Due to the specialisation of these projects and their lack of separation between the concerns of engagement design and core infrastructure, in practice any context which requires an engagement archetype that significantly differs from a talk show necessarily requires the design and implementation of a new, custom-built platform.

2.3 Ontologies, Design Vocabularies, & Group Telephony

Having an agreed-upon terminology to describe the components, behaviours, and characteristics of emerging or underexplored technologies can support the design, development and comparison of implementations of them. Formal ontologies are often produced with this use-case in mind: intended to create '*a shared and common understanding of some domain that can be communicated across people and computers*' [33]. They can be used to create and communicate consensual conceptual models (referred to as 'domain reference' ontologies), which can then be further developed to be used to support semantic interoperability within automated systems (referred to as being 'operational') [13] when written in a machine-interpretable language such as OWL (Web Ontology Language[2]). To support extensibility and reusability, ontologies are frequently organised within three tiers: *foundational*, which describe fundamental concepts that can be shared and reused across many fields (e.g. 'a thing'); *core*, which build upon foundational axioms to provide abstract definitions more specific to a particular field (e.g. 'a document is a thing'); and *domain*, a definition of concepts and relations concerning a specific, specialised domain (e.g. 'a forum post is a document') [29]. Costa et al. note that ontologies have been used to structure and infer knowledge, support reasoning, and serve as a basis for developing systems or frameworks across many portions of the HCI domain—such as within user interface design, ubiquitous computing, user modeling, design methods, usability, and accessibility [6].

Beyond formal ontologies, other approaches have been used within HCI to describe and illustrate the characteristics of technologies and interactions with them. For example, Kharrufa et al. contributed a conceptual model which allowed for the visual communication of different approaches to user identification on multi-touch devices: aiming to support non-technical practitioners to compare the common parameters that characterize them, independent of their specific use-cases [17]. Similarly, Chuang et al. produced a 'design vocabulary' to describe user-computer communication within IoT systems, supporting the comparison and design of different forms of interaction [4]. In comparison to the aforementioned machine-interpretable ontologies which focus on describing existing implementation, these examples instead place greater focus on future design: communicating a system's requirements, strengths, limitations, or possibility space through the use of visuals and hypothesised future technologies.

While prior works have explored the use of ontologies to drive contextual dialogues in IVR systems [1, 36, 37], detail the technical aspects of radio transmission and operation [14], describe media industry production workflows [21], and support the automatic processing and annotation

[2] https://www.w3.org/OWL/

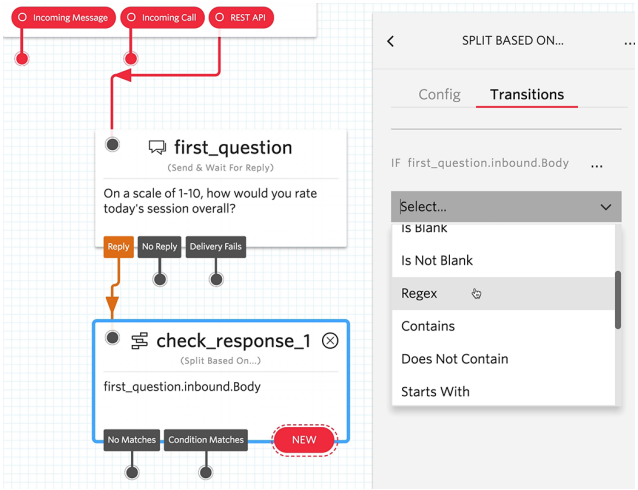

Fig. 1. Twilio's Studio interface, which allows for producing IVR logic through a visual programming interface.

of meetings [23], we have not identified an ontology or design vocabulary that describes the configuration and composition of synchronous group interaction archetypes, or how they can be scaffolded or facilitated remotely through telephony. Tools do exist to rapidly develop (and therefore describe) IVR platforms using visual interfaces (e.g. Twilio Studio[3]), allowing users to scaffold IVR interactions without requiring significant programming experience through a visual representation of logic (Figure 1). However, these tools largely focus on the design of asynchronous interactions (e.g. SMS surveys, IVR menus) or connecting callers to separately handled synchronous activities (e.g. a helpline), without describing or offering granular controls over synchronous group interactions. As such, there exists a need for a set of agreed upon terms and metaphors to assist designers in conceptualising, visualising, and communicating synchronous group telephony systems and interactions.

3 ONTOLOGY DEVELOPMENT

In response, we set out to develop a vocabulary which could be used by designers and developers as a tool to easily design and communicate different models of synchronous audio engagements, describe existing technologies, and highlight other novel ways in which such platforms could be used. As such, we decided to produce two outputs: a formal ontology called Scaffolding Interactions through Synchronous Telephony (SIST), which would define the terms for the necessary system components and how they interact; and a design vocabulary built upon SIST, supporting less technical stakeholders in exploring potential designs and identifying their strengths, weaknesses and requirements through a visual format reminiscent of Twilio's Studio interface.

To produce SIST, we followed the NeOn methodology for developing ontologies [12, 34], which involves specifying: the aims of the ontology and its intended users; the functional requirements it should fulfill through the generation of competency questions; what existing ontological resources could be expanded upon or repurposed; and the process through which the ontology should be developed. This section describes the outputs of this methodology.

[3]https://www.twilio.com/serverless/studio

3.1 Requirements Specification

We aimed to produce an ontology describing the core components common to existing synchronous telephony platforms, as well as additional concepts that would allow a platform to dynamically support different formats of group interactions. This required analysing the established design patterns (i.e. the talk show format) and functionality used by previous projects in the space [16, 35, 42].

3.1.1 The talk show format. In existing platforms, participants join an engagement by either receiving a phone call or initiating a 'callback' (where the system hangs up on incoming calls and immediately calls them back, so that participants aren't charged). Participants spend the majority of the engagement muted, passively listening to one or more unmuted users: a 'host' user and, if present, guests (typically experts in their field, e.g. healthcare providers [35]). These discussions are typically unscripted. The engagements often also have a question and answer (Q&A) segment, where members of the audience can request to be unmuted by pressing a button on their phone. The host can see these requests through a visual interface, and choose who to unmute and when to re-mute them. The audience member's question would then be discussed by the host and/or guests before moving onto the next question. Some platforms also support the recording and playback of audio during the call [35, 42].

3.1.2 Utilised functionality. To support the talk show format, these platforms use many of the features made readily available by software such as FreeSWITCH: dialing and disconnecting callers; muting and unmuting individuals; transferring callers into a single 'conference' group call; and recording and playing back audio files—either to individuals or everyone on the call. To be of use, these functions need to be mediated by an application with which the host can interact. Our ontology would need to describe these functions, as well as how they could be combined and utilised either programmatically or manually by the host to support different engagement formats.

3.1.3 Intended use. The intended users of SIST are those who would want to explore the synchronous group telephony design space by implementing a platform which combined these functionalities to create new engagement formats and experiences. For example, we were interested in exploring if multiple simultaneous 'conference calls' could be used to replicate Zoom's 'breakout room' functionality. To support this intended use, we decided to produce SIST as a domain reference ontology in OWL: featuring enough detail and scope to adequately describe the key components of a platform's required architecture and engagements held through it, but not so much to as inhibit design explorations.

NeOn recommends generating generating competency questions which you would want the ontology to be able to address. Ours included: *CQ1: 'What functionality is required to support a breakout room activity?'*, *CQ2: 'Which participants' questions were unaddressed in the last engagement?'*, and *CQ3: 'How can a pre-show segment be structured with automated playback of music and announcements?'*.

3.2 Reusing Ontological Resources

NeOn also recommends researching which existing ontological resources could be adapted or expanded upon, which can both avoid 'reinventing the wheel' [34] and support extensibility and use alongside other vocabularies within a network by avoiding the introduction of new core concepts [29]. All of SIST's axioms are built upon two previously established ontological resources: Friend of a Friend (aka '*foaf*': an ontology describing people, their activities and their relations to other people

and objects[4]) and Semantically Interlinked Online Communities (aka '*sioc*': an ontology which builds on *foaf* to describe online community spaces, such as forums[5]). These existing ontologies were deemed suitable to be used as core vocabularies from which to develop SIST, as they are already used to describe the configurations of virtual discussion spaces (e.g. the axioms *sioc:Space*, *sioc:Site*, *sioc:Container*), the users and systems who interact with them (*foaf:Agent*, *sioc:UserAccount*, *sioc:Usergroup*), and the data produced by these interactions (*foaf:Document*, *sioc:Item*).

3.3 Iterative-Incremental Development Cycle

As we aimed for SIST to describe an underexplored domain with uncertain requirements, we chose to follow the iterative development cycle recommended within the NeOn methodology. SIST went through three major revisions of development by this paper's first two authors: each revision building towards fulfilling a greater number of competency questions, before undergoing review with the other authors. To facilitate communication with authors unfamiliar with OWL, these reviews utilised visual representations of SIST generated with draw.io[6] (as shown in Figure 2). For the sake of clarity and brevity, these diagrams excluded the base-level *foaf* and *sioc* axioms (the full class hierarchy is visible in Figure 3).

4 ONTOLOGY OVERVIEW

Made up of relatively few classes, SIST is designed to describe and structure the design space of synchronous group telephony engagement platforms, whilst still remaining abstract enough to encourage exploration, interpretation and expansion. The ontology can be conceptualised as being made up of five layers of components: *UI*, which describes how users and hosts interact with the system through phone calls and visual applications; *Scaffolding*, which defines the components that, when combined, create the logic to orchestrates different engagement formats; *System*, which describes the software (such as FreeSWITCH) by which the phone calls are initiated and manipulated; *App*, which manages and enacts the Scaffolding logic and directs the System layer components; and *Data*, which refers to the data created, stored and retrieved by the App layer.

Two forms of visual representation were also developed: one to communicate the relationships between the different layers of components of the ontology (Figure 2); and the SIST visual design vocabulary, which aims to support designers in designing, communicating, and understanding the technical requirements of different types of group engagements over synchronous telephony (an example is given in Figure 4). This section will give a detailed overview of the SIST ontology, using these visuals as a communication aid.

4.1 The UI Layer

The UI Layer describes users and the methods by which they interact with the system—be that by a phone call or through a visual interface. Users (instances of *foaf:Person*) interacting with the system are either a Host or Member (subclasses of *sioc:UserAccount*).

Members are regular participants, and engage with the system through PhoneCalls (extends *sioc:Space*). Through PhoneCalls, they encounter the system by populating audio spaces referred to as Rooms (extends *sioc:Container*). Prior platforms only utilised a single Room, where all users were in a single conference call, but (as discussed later) multiple could be created simultaneously. MemberInteractions (extending *sioc:Item*) occur inside these rooms, and could be utterances of speech or button presses.

[4] http://xmlns.com/foaf/0.1/
[5] http://rdfs.org/sioc/spec/
[6] https://app.diagrams.net/

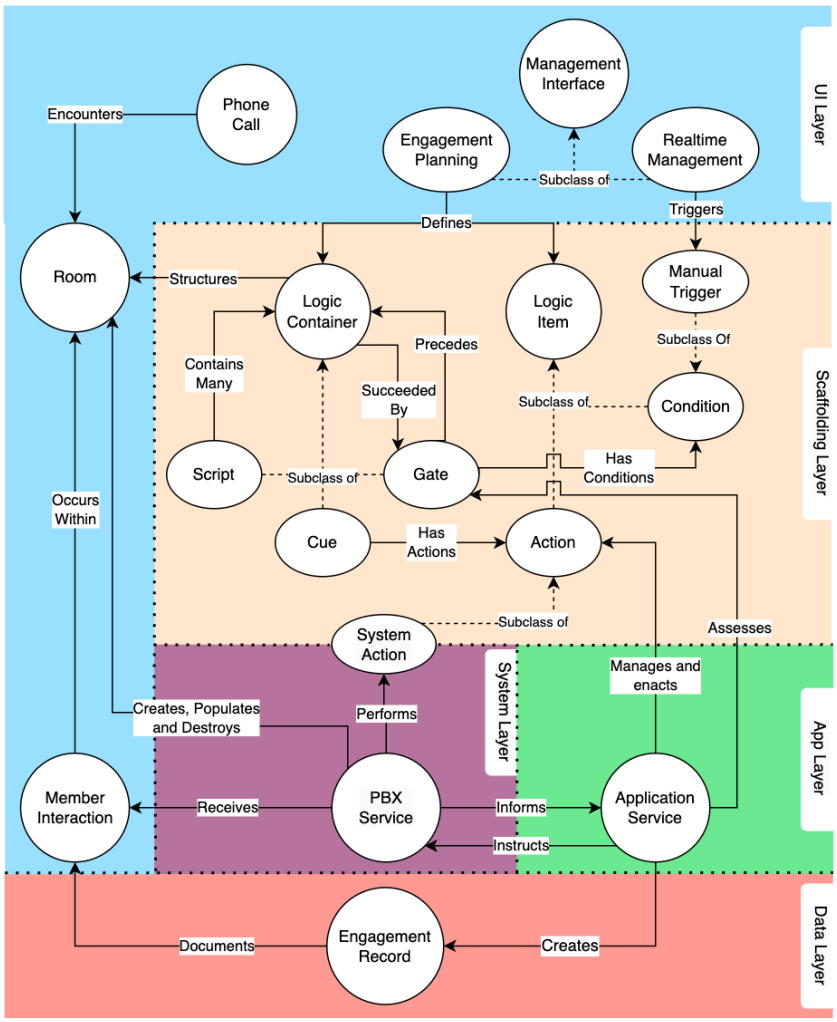

Fig. 2. A visualisation of the classes and their relationships within SIST, abbreviated for clarity. Solid arrows represent object properties (relationships). Circles depict classes whose parent classes are not shown (all are descendants from *foaf* or *sioc* classes). Ovals represent subclasses, connected to their parent class by dashed arrows (e.g. SystemAction is a subclass of Action).

Hosts can access the visual ManagementInterface (extends *sioc:Site*), which has two implementations depending on the state of the call: EngagementPlanning if the call is yet to start, and RealtimeManagement if it has. As an example, the Android app in [16] is a ManagementInterface, used during the call to keep track of which participants wanted to ask questions, and mute and unmute them as needed.

4.2 The Scaffolding Layer

The interplay of the components of the Scaffolding Layer creates the blueprints for different structured interactions (e.g. a Q&A section, a breakout room session) during a call. These components are defined by the host through the EngagementPlanning interface. The permissions a participant has

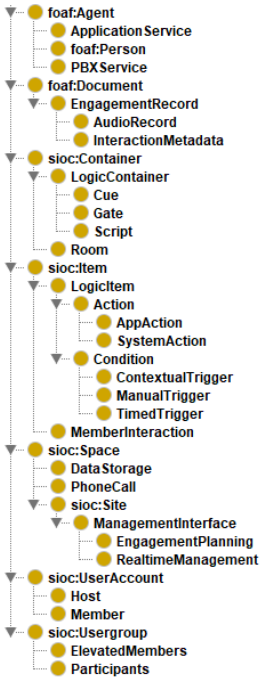

Fig. 3. Hierarchy of the classes utilised by SIST, as displayed in the OWL development application Protégé.

(e.g. if they can unmute themselves) can be controlled by assigning them to one of two subclasses of *sioc:Usergroup*: ElevatedMembers or Participants (omitted from Figure 2 for space reasons). The remaining components are either LogicContainers (subclass of *sioc:Container*) or LogicItems (subclass of *sioc:Item*). LogicContainers contain LogicItems, and are assessed and enacted by the ApplicationService sequentially (similar to Twilio Studio). To support synchronous interactions and branching logic, multiple LogicContainers can be active at the same time (as demonstrated in Figure 5). A LogicContainer can be either a Script, Cue or Gate, whereas LogicItems are Actions or Conditions:

4.2.1 Condition (LogicItem). A small piece of logic relating to the call that can be evaluated to a Boolean (true or false) value. There are three main types of Conditions: TimedTriggers, which evaluate to true after a given length of time; ManualTriggers, which only become true through direct intervention by the host through the RealtimeManagement interface (e.g. the host presses a button to play an audio file); and ContextualTriggers, which can relate to particular factors within the call (e.g. 'there are 10 participants present', 'the audio file has finished playback'). ContextualTriggers are less formally defined in the vocabulary to support application-specific implementation.

4.2.2 Action (LogicItem). An active intervention which can affect the flow of the call, managed by the ApplicationService in the Application Layer. Actions are one of two types: SystemActions and AppActions. SystemActions are performed in the System Layer as they directly interact with call audio or connections. They can be one of a limited set of interactions: *dial*, *hangup*, *mute*, *unmute*, *transfer*, *createRoom*, *closeRoom*, *playAudioToMember*, *playAudioToRoom*, *stopPlayback*, *startRecording*, and *stopRecording*. As with ContextualTriggers, AppActions are less strictly defined as they likely require per-application implementation within the ApplicationService. Examples

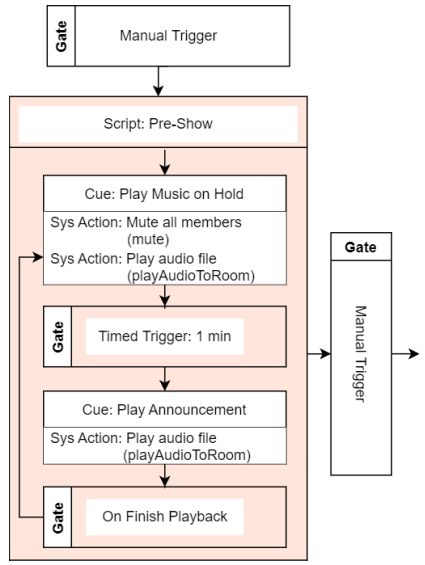

Fig. 4. Using the visual design vocabulary to show how elements of the Scaffolding Layer can be configured to produce an automated pre-show routine. Here, the 'Pre-Show' Script is initialised by the Host through a ManualTrigger. The Script contains four LogicContainers: two Gates and two Cues. The second Gate precedes the first Cue, forming a logic loop. This loop can be interrupted by the Host through a ManualTrigger, which would exit the Script.

include: 'Mark participant Y as having raised their hand', 'Move to the next agenda item', 'Make participant X an ElevatedMember'.

The functionality required for an interaction's implementation can be identified by examining what LogicItems it uses: for example, listing the Conditions and Actions used in Figure 6 provides an answer to *CQ1*.

4.2.3 Gate (LogicContainer). Gates precede (and can succeed) other LogicContainers. They contain one or more Conditions connected by the logic operators AND or OR, and 'unlock' once the ApplicationService in the System Layer evaluates the expression as true. For example: 'The participant presses 8 AND the participant is muted'. Once the Gate is 'unlocked', the LogicContainer it precedes becomes active. To support branching logic, multiple parallel Gates can be assessed simultaneously, each preceding a different LogicContainer (e.g. multiple options in an IVR menu: '*press 1 to do X, press 2 to do Y*').

4.2.4 Cue (LogicContainer). Cues contain one or more Actions which are enacted when the preceding Gate is unlocked. If a succeeding Gate exists, it will start being assessed once the Cue's Actions have been initiated.

4.2.5 Script (LogicContainer). Scripts contain other LogicContainers (including other Scripts) to support containerisation: allowing extended logic sequences to be initiated or interrupted by Gates. As an example, Figure 4 uses the visual design vocabulary to show a Script for a looping pre-show routine. In the routine, participants in the call are muted and music plays. After one minute, a pre-recorded announcement plays, and after it finishes the music plays again. As this Script is a LogicContainer with a succeeding Gate, the call's host can interrupt this loop at any time to start

the show. If no loop is present, the final LogicContainer will exit the script upon completion (as seen in Figures 6 and 7).

As will be demonstrated in Section 5, through such combinations of Conditions, Gates, Actions, Cues, and Scripts, we can use SIST to describe complex interactions.

4.3 The Application Layer

The Application Layer contains the ApplicationService: a software application (e.g. built using PHP, as in [16]) which puts the Scaffolding Layer into practice, assessing currently active Gates and managing any resulting Actions. It enacts all except SystemActions, which it instructs the System Layer (described below) to perform. The ApplicationService is a *foaf:Agent*, as it is capable of programmatically taking independent action in response to changes in state.

4.4 The System Layer

The System Layer features the PBXService (also a *foaf:Agent*): the key server-side component common to previous synchronous telephony platforms [16, 35, 42]. PBXService is an application capable of programmatically initiating and managing voice connections (e.g. via Voice Over IP, or the Public Switched Telephone Network) and creating, populating and destroying Rooms. FreeSWITCH is an example of a PBXService, and was used by Kazakos et al. in their project [16]. The PBXService enacts SystemActions and informs the ApplicationService of any changes in the call state, such as instances of MemberInteractions (e.g. buttons presses, vocal utterances, disconnections).

4.5 The Data Layer

The Data Layer concerns the storage of data, either for use within a call (e.g. playback of existing audio) or archiving. EngagementRecords (subclass of *foaf:Document*) can either be AudioRecords or InteractionMetadata, and are created by the ApplicationService to document MemberInteractions. As an example of how these could be used, the process described in Figure 5 could keep records of which Members raised their hands and which were unmuted—data which could be queried to answer *CQ2*.

5 SCAFFOLDING ENGAGEMENTS THROUGH SIST'S DESIGN VOCABULARY

This section will use the visual design vocabulary to describe and demonstrate the use of SIST with three different archetypes of group interactions: talk shows, research workshops, and speed dating events.

5.1 Scenario 1: Talk Show

Figure 5 shows how a talk show engagement format (described in Section 3.1.1) with strict time-keeping can be scaffolded programmatically using SIST. The main segments of the show (pre-show, guest discussion, and audience Q&A) are transitioned between by the host through the use of Manual Triggers. The use of containerisation means that the 'pre-show' Script previously shown in Figure 4 can be re-used. The 'guest discussion' unmutes the Elevated Members group (i.e. the host and guests), and mutes everyone else. The 'audience Q&A' Script has two sets of asynchronous tracks: one which is used on a per-Member basis, letting callers add and remove themselves from the queue to ask questions through a button press; and one for the host to manage the queue. When triggered by the host, the Member at the front of the queue will be unmuted for 30 seconds (or manually interrupted by the host) to ask a question, before being muted again and removed from the queue. The Script is exited by a Manual Trigger, after which further logic could either end the show, or clear the queue and ensure that all participants are muted again before continuing. Planning the interaction through SIST and observing used LogicItems shows that this system would

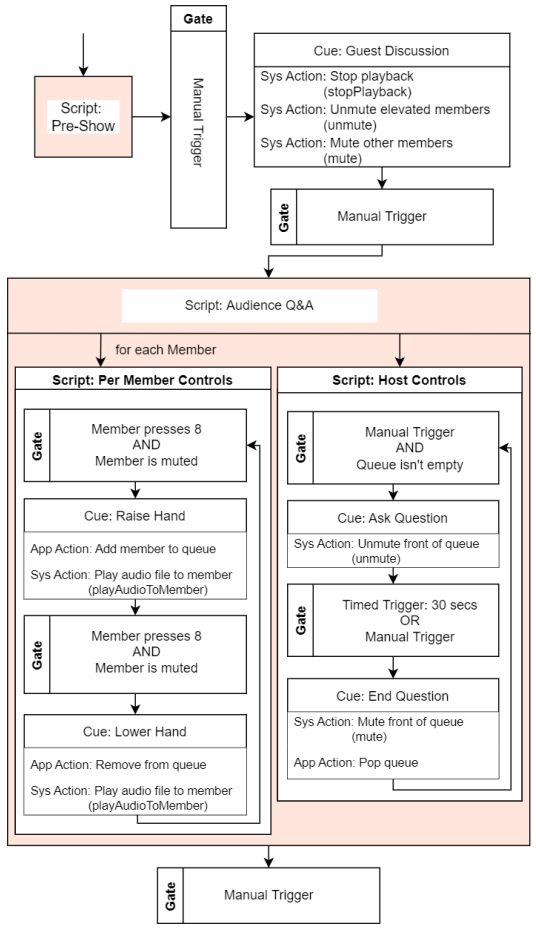

Fig. 5. Using SIST to prototype the interactions involved in a talk show with an audience Q&A segment. The 'Pre-Show' Script is detailed in Figure 4. During the Q&A, multiple Scripts are active at once: one for the Host, and one for each Member. Note that the handling of race conditions has been omitted to save space.

need to implement mute, unmute, and audio playback SystemActions, as well as a queue system which can be manipulated by callers and the host.

5.2 Scenario 2: Research Workshop

5.2.1 Overview. Since the start of the COVID-19 pandemic, it has become more common to hold research workshops remotely though platforms such as Zoom: the 'breakout room' feature (where callers can be temporarily subdivided into smaller discussion groups) can be used as an analogue for having separate tables of participants within a qualitative workshop to support group discussion and activities. This degree of dynamic manipulation of call topology has not yet been supported within existing synchronous group telephony platforms.

5.2.2 SIST Implementation. Figure 6 shows how SIST could be used to scaffold a 'breakout' segment of a research workshop though synchronous telephony (answering *CQ3*). Prior to commencing, this Script requires a configuration noting the number of Rooms to create, the length of time the

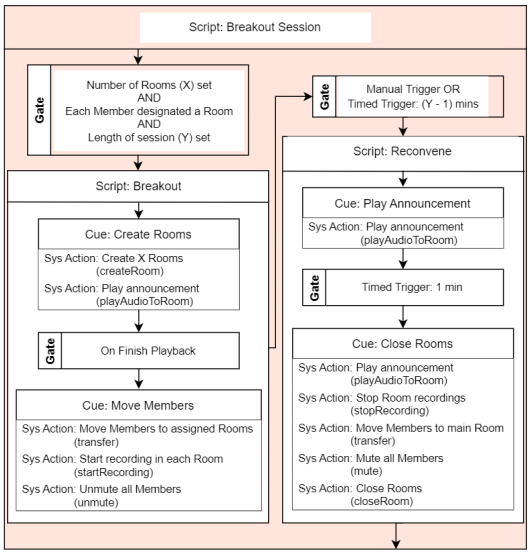

Fig. 6. Using SIST to scaffold a 'breakout room' interaction to be used during a distributed research workshop

breakout session should run for, and an assignment of which Members should go to each room. The Script will create the given number of Rooms, play a pre-recorded announcement to Members, and then move them to their assigned Room. The system records each room to a separate audio file for later reference by the researchers, who may not be present in all Rooms. When there is one minute left (or the Host intervenes) another announcement is played, and the Members are muted and moved back to the main Room when the timer expires. This interaction requires the PBXService to be able to create Rooms and assign Members to them: within FreeSWITCH (used by [16, 35, 42]), this can be achieved through the use of conferences[7] and transfers[8].

5.3 Scenario 3: Speed dating

5.3.1 Overview. Speed dating events typically involve a group of individuals being split into pairs to have short, one-on-one conversations, with pairs cycling every few minutes so that all suitably matched participants have a chance to meet. While fairly simple to organise in person, doing such structured interactions remotely through standard telephony would be highly labour intensive and prone to error: likely requiring participants manage their own timings and hold separate phone calls for each pairing.

5.3.2 SIST Implementation. Figure 7 shows the 'speed dating' example, and demonstrates how programmatically manipulating the topology of meeting rooms opens up many new opportunities for remote interaction. This design uses the previous breakout room Script (altered to disable recording functionality) to facilitate the one-on-one conversations, with each pairing having their own Room. While the previous example relied on the Host manually assigning participants to groups, here the pairings would be assigned through the use of a round-robin algorithm[9], managed by the ApplicationService.

[7] https://freeswitch.org/confluence/display/FREESWITCH/mod_conference
[8] https://freeswitch.org/confluence/display/FREESWITCH/mod_dptools%3A+transfer
[9] https://en.wikipedia.org/wiki/Round-robin_tournament

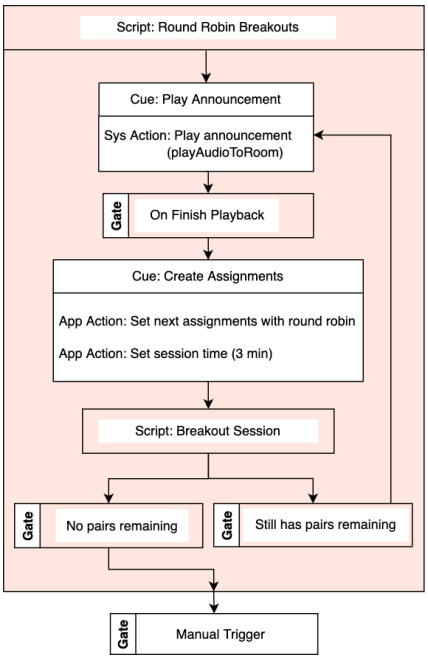

Fig. 7. Using SIST to scaffold a 'speed dating' interaction using breakout rooms in a round robin configuration. The 'Breakout Session' Script can be re-used from Figure 6 with minimal modification.

6 DISCUSSION

6.1 Supporting Design Exploration and Sustainability through Modularity

A key feature of SIST is its separation of a platform's component parts into five distinct layers: *UI*, *Scaffolding*, *Application*, *System* and *Data*. By separating a synchronous telephony platform's implementation out into these layers, SIST allows for the partial decoupling of the systems' key components from each other. A key benefit of this separation of concerns is reusability: different combinations of interactions can be implemented in the scaffolding layer, often without the need to update the rest of the system.

This separation is particularly beneficial to the System Layer: while PBXServices such as FreeSWITCH are freely available, installing and configuring them can be complicated and time consuming. Having such a limited number of SystemActions which can be performed by the PBXService provides the System Layer with a core feature-set: one which is unchanging between applications, but can be combined with logic from the Scaffolding and Application Layers to support a wide variety of interaction archetypes. For example, the research workshop and speed dating scenarios each re-used the same SystemActions ('*createRoom*' and '*transfer*') to produce markedly different experiences, without the need to change the PBXService.

In contrast, existing synchronous group telephony platforms (as described in their publications) seem to have been explicitly designed to implement a single engagement format [16, 35, 41]. As such, the implementation of these platforms is tightly coupled to the 'talk show' format, and implementing new interaction methods would require significant re-engineering. While introducing new AppActions in SIST will likely always require the development of additional features within

the ApplicationService (e.g. the production of a queue system, or an implementation of round-robin), this can still be achieved without touching the PBXService: meaning that developers of such features don't necessarily need to deploy or have a technical understanding of complicated platforms like FreeSWITCH. As a result, we argue that the reusability supported by an approach like SIST results in a much lower barrier to the design and prototyping of platforms for synchronous group telephony.

6.2 Scaffolding New Interactions through Synchronous Group Telephony

While the talk show format supported by previous synchronous telephony projects can be effective at delivering information to groups of participants, it offers limited opportunities for participant interaction and agency as, by necessity, participants spend most of the call muted. Not only is this incongruous with most participants' prior experiences of phone calls (which typically consist of unstructured discussion between two individuals), but it is also incompatible with any use-cases which require open discussion between participants, or for all participants (including the host) to be on equal footing. Despite these limitations, the use of alternatives to the talk show format within synchronous group telephony has not been adequately explored. As discussed, we believe that this is partially because prior platforms seem to have been designed around talk shows, and would require significant re-engineering to test new formats or be adapted in response to changing stakeholder requirements. However, we also posit that the lack of well defined and commonly agreed axioms can make designing, communicating, and visualising the processes of these engagement formats unwieldy.

We argue that the example scenarios presented in this paper using the visual vocabulary demonstrate that SIST is flexible enough to support the exploration, development and communication of a wide variety of group interaction archetypes through synchronous telephony. The use of SIST to scaffold a talk show format (Figure 5), complete with an automated pre-show routine (Figure 4), demonstrates that it can be used to describe the engagements facilitated by existing synchronous group telephony platforms, and further highlights the value of a vocabulary which is able to describe and communicate these complex systems. Furthermore, during our experiences developing these scenarios through the design vocabulary there were multiple points where we had to consider unanticipated system requirements, or gaps in logic. For example, when defining the timed Gate to reconvene breakout groups (Figure 6), we realised that there should be a manual override available to the host in case they wanted to reconvene the groups early. Such instances highlight that this format has value in being used for developing low-fidelity logical prototypes.

This paper has also demonstrated how the Scaffolding and App Layers can be used to produce new engagement models, beyond the capabilities of previous synchronous group telephony platforms. For example, while prior platforms limited participants to the largely passive role of an audience member, the research workshop example (Figure 6) supports all participants engaging in open discussion through the use of breakout rooms, without requiring anyone to be muted. The speed dating example (Figure 7) demonstrates how this could be taken further: by implementing the 'round-robin' algorithm in the ApplicationService to determine participant pairings, it demonstrated the potential for programmatic scaffolding to produce remote encounters where all participants have comparable levels of agency. This could be expanded even further to support highly structured engagement formats, such as Robert's Rules of Order: a parliamentary procedure used in many contexts to support group discussion and decision making with '*due regard for every member's opinion*' [27]. In such a case, the ApplicationService acts as the arbiter of the engagement—able to direct proceedings, potentially with minimal intervention from the Host.

Such possibilities suggest that synchronous telephony offers potential for facilitating group interactions beyond those covered in the scenarios presented in this paper. We argue that this

potential can be more easily explored through SIST: that the visual design vocabulary supports the design, communication, and analysis of a wide array of new engagement formats hitherto unutilised in remote contexts with offline participants.

6.3 Accessing Under-Explored Design Spaces

One of the key benefits of a common vocabulary is that it can be used to share an understanding of a domain which can be communicated across people, computers or both [33]. As SIST's design vocabulary is more approachable, less technical, and more easily communicable than the underlying architectures and systems it describes, it offers designers and engineers a bridge for discussion and exploration of the potential interactions offered by synchronous group telephony. Furthermore, as this visual format is grounded by a formal ontology, it opens a design space that can be referenced, shared, and built upon by different systems—independent from their underlying software stack. Such an approach lowers the barrier to entry to the exploration, experimentation and appropriation of the uses of synchronous group telephony in new engagement contexts and use cases.

The layered framing of SIST also supports approaching each of these layers as their own design spaces, independent of the rest of the platform's infrastructure. The Data Layer provides a strong example: while explorations have been made regarding how systems can support participatory recordkeeping practices [28], the HCI community has performed little to no research into how organisations performing voice-based stakeholder engagements can (or should) approach record-keeping practices. Researchers within this design space could explore how informed consent should be taken, recorded or rescinded through voice-based platforms; or how resulting audio records could be created, stored, accessed, documented, appraised, and disposed of through voice-based interfaces. Examples of what explorations could be made within the Application Layer include investigating how programmatic, agent-based systems (such as chatbots [15, 41] or virtual participants [2]) could be incorporated into the synchronous activities offered by these platforms. As such, we posit that the layers of SIST are rich design spaces, within which we can explore how other practices and technologies can interact with the medium of synchronous group telephony.

7 FUTURE WORK

We are currently in the process of incorporating SIST into a demonstrable synchronous telephony platform. Within our future work, we plan to use SIST to support the kinds of group interaction formats illustrated in this paper's scenarios, explore some of the previously discussed record-keeping concepts within the Data Layer, and investigate how SIST could also be used to scaffold asynchronous interactions before, during and after synchronous engagements [42]. Through the implementation of this platform, we hope to be able to provide other researchers in this space recommendations for how to put SIST into practice, as well as further refine the vocabulary in response to findings resulting from its implementation.

It is also worth noting that much of SIST could be applied to other mediums, such as synchronous video. We wished to focus on its application to group telephony within this paper, given that it is a particularly under-explored design space. However, as SIST is a formal ontology, future work could extend it to suit other mediums.

8 CONCLUSION

Recent years have demonstrated the value and potential of synchronous telephony platforms which enable engagements with remote, offline groups of participants through traditional phone networks. However, this potential has been under-explored, with all published platforms in this space following the same 'talk show' engagement format. In response, this paper presents SIST: a design vocabulary, grounded in a formal ontology describing the components of a platform

needed to support the dynamic scaffolding of group interactions through synchronous telephony. Demonstrating through a series of scenarios describing three different use cases of synchronous group telephony, we argue that SIST is not only capable of describing the interactions enabled by previously published systems, but also new interaction formats with a flexibility previously unavailable within remote telephony. We argue that SIST provides an approachable, structured vocabulary which offers a more sustainable framework for system design, and highlights that the layers of components within these systems each offer a rich, under-explored design space.

ACKNOWLEDGMENTS

REFERENCES

[1] Mohammad Ababneh and Duminda Wijesekera. 2013. An Ontological Inference Driven Interactive Voice Response System. *Processing of Semantic Technology for Intelligence, Defense, and Security (STIDS), USA* (2013), 125–132.

[2] Tom Bartindale, Peter Chen, Harrison Marshall, Stanislav Pozdniakov, and Dan Richardson. 2021. ZoomSense: A Scalable Infrastructure for Augmenting Zoom *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 3771–3774. https://doi.org/10.1145/3474085.3478332

[3] Nicola J. Bidwell, Roberto Cibin, Conor Linehan, Laura Maye, and Sarah Robinson. 2021. Being Regulated: Licence to Imagine New Technology for Community Radio. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 154 (apr 2021), 27 pages. https://doi.org/10.1145/3449228

[4] Yaliang Chuang, Lin-Lin Chen, and Yoga Liu. 2018. Design Vocabulary for Human–IoT Systems Communication. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173848

[5] Roberto Cibin, Maurizio Teli, and Sarah Robinson. 2019. Institutioning and Community Radio. A Comparative Perspective. In *Proceedings of the 9th International Conference on Communities & Technologies - Transforming Communities* (Vienna, Austria) *(C&T '19)*. Association for Computing Machinery, New York, NY, USA, 143–154. https://doi.org/10.1145/3328320.3328392

[6] Simone Dornelas Costa, Monalessa Perini Barcellos, and Ricardo de Almeida Falbo. 2021. Ontologies in Human–Computer Interaction: A Systematic Literature Review. *Appl. Ontol.* 16, 4 (jan 2021), 421–452. https://doi.org/10.3233/AO-210255

[7] Chris Csíkszentmihályi and Jude Mukundane. 2016. RootIO: ICT + telephony for grassroots radio. In *2016 IST-Africa Week Conference*. 1–13. https://doi.org/10.1109/ISTAFRICA.2016.7530700

[8] Bronwyn J. Cumbo, Tom Bartindale, and Dan Richardson. 2021. Exploring the Opportunities for Online Learning Platforms to Support the Emergency Home School Context. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 347, 11 pages. https://doi.org/10.1145/3411764.3445044

[9] Anton Eitzinger, James Cock, Karl Atzmanstorfer, Claudia R. Binder, Peter Läderach, Osana Bonilla-Findji, Mona Bartling, Caroline Mwongera, Leo Zurita, and Andy Jarvis. 2019. GeoFarmer: A monitoring and feedback system for agricultural development projects. *Computers and Electronics in Agriculture* 158 (2019), 109–121. https://doi.org/10.1016/j.compag.2019.01.049

[10] Hira Ejaz, Syed Ali Hussain, and Agha Ali Raza. 2018. The Case for IVR-Based Citizen Journalism in Pakistan. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (Barcelona, Spain) *(MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, 87–94. https://doi.org/10.1145/3236112.3236124

[11] Casey Fiesler, Jed R. Brubaker, Andrea Forte, Shion Guha, Nora McDonald, and Michael Muller. 2019. Qualitative Methods for CSCW: Challenges and Opportunities. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing* (Austin, TX, USA) *(CSCW '19)*. Association for Computing Machinery, New York, NY, USA, 455–460. https://doi.org/10.1145/3311957.3359428

[12] A. Gómez-Pérez and Mari Carmen Suárez-Figueroa. 2009. NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology. In *Proceedings of International Conference on Software, Services & Semantic technologies (S3T 2009)*. Universidad de Sofia, Bulgaria. https://oa.upm.es/5475/

[13] Giancarlo Guizzardi. 2007. On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In *Proceedings of the 2007 Conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference DB&IS'2006*. IOS Press, NLD, 18–39.

[14] IHMC. [n.d.]. IHMC Ontology and Policy Management: Radio Ontology — ontology.ihmc.us. https://ontology.ihmc.us/Radio/index.html. [Accessed 19-Aug-2022].

[15] Mohit Jain, Pratyush Kumar, Ishita Bhansali, Q. Vera Liao, Khai Truong, and Shwetak Patel. 2018. FarmChat: A Conversational Agent to Answer Farmer Queries. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 170 (dec 2018), 22 pages. https://doi.org/10.1145/3287048

[16] Konstantinos Kazakos, Siddhartha Asthana, Madeline Balaam, Mona Duggal, Amey Holden, Limalemla Jamir, Nanda Kishore Kannuri, Saurabh Kumar, Amarendar Reddy Manindla, Subhashini Arcot Manikam, GVS Murthy, Papreen Nahar, Peter Phillimore, Shreyaswi Sathyanath, Pushpendra Singh, Meenu Singh, Pete Wright, Deepika Yadav, and Patrick Olivier. 2016. A Real-Time IVR Platform for Community Radio. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 343–354. https://doi.org/10.1145/2858036.2858585

[17] Ahmed Kharrufa, Thomas Ploetz, and Patrick Olivier. 2017. A Unified Model for User Identification on Multi-Touch Surfaces: A Survey and Meta-Analysis. *ACM Trans. Comput.-Hum. Interact.* 24, 6, Article 39 (dec 2017), 39 pages. https://doi.org/10.1145/3144569

[18] Aman Khullar, Priyadarshi Hitesh, Shoaib Rahman, Deepak Kumar, Rachit Pandey, Praveen Kumar, Rajeshwari Tripathi, Prince Prince, Ankit Akash Jha, Himanshu Himanshu, and Aaditeshwar Seth. 2021. Costs and Benefits of Conducting Voice-Based Surveys Versus Keypress-Based Surveys on Interactive Voice Response Systems. In *ACM SIGCAS Conference on Computing and Sustainable Societies* (Virtual Event, Australia) *(COMPASS '21)*. Association for Computing Machinery, New York, NY, USA, 288–298. https://doi.org/10.1145/3460112.3471963

[19] Aman Khullar, M Santosh, Praveen Kumar, Shoaib Rahman, Rajeshwari Tripathi, Deepak Kumar, Sangeeta Saini, Rachit Pandey, and Aaditeshwar Seth. 2021. Early Results from Automating Voice-Based Question-Answering Services Among Low-Income Populations in India. In *ACM SIGCAS Conference on Computing and Sustainable Societies* (Virtual Event, Australia) *(COMPASS '21)*. Association for Computing Machinery, New York, NY, USA, 79–87. https://doi.org/10.1145/3460112.3471946

[20] Laura Maye, Sarah Robinson, Nadia Pantidi, Liana Ganea, Oana Ganea, Conor Linehan, and John McCarthy. 2020. *Considerations for Implementing Technology to Support Community Radio in Rural Communities.* Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376580

[21] MovieLabs. 2021. MovieLabs Publishes Ontology for Media Creation - MovieLabs — movielabs.com. https://movielabs.com/news/movielabs-publishes-ontology-for-media-creation/. [Accessed 19-Aug-2022].

[22] Tembalethu Jama Ndwe, Etienne Barnard, and Mqhele E. Dlodlo. 2012. Effects of Application Type on the Choice of Interaction Modality in IVR Systems. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (Pretoria, South Africa) *(SAICSIT '12)*. Association for Computing Machinery, New York, NY, USA, 236–242. https://doi.org/10.1145/2389836.2389864

[23] John Niekrasz and Matthew Purver. 2006. A Multimodal Discourse Ontology for Meeting Understanding. In *Machine Learning for Multimodal Interaction*, Steve Renals and Samy Bengio (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 162–173.

[24] Neil Patel, Deepti Chittamuru, Anupam Jain, Paresh Dave, and Tapan S. Parikh. 2010. Avaaj Otalo: A Field Study of an Interactive Voice Forum for Small Farmers in Rural India. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 733–742. https://doi.org/10.1145/1753326.1753434

[25] Waleed Riaz, Haris Durrani, Suleman Shahid, and Agha Ali Raza. 2017. ICT Intervention for Agriculture Development: Designing an IVR System for Farmers in Pakistan. In *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development* (Lahore, Pakistan) *(ICTD '17)*. Association for Computing Machinery, New York, NY, USA, Article 33, 5 pages. https://doi.org/10.1145/3136560.3136598

[26] Dan Richardson, Bronwyn J. Cumbo, Tom Bartindale, Delvin Varghese, Manika Saha, Pratyasha Saha, Syed Ishtiaque Ahmed, Gillian C. Oliver, and Patrick Olivier. 2022. Critically Engaging with Embedded Values through Constrained Technology Design. In *Designing Interactive Systems Conference* (Virtual Event, Australia) *(DIS '22)*. Association for Computing Machinery, New York, NY, USA, 643–653. https://doi.org/10.1145/3532106.3533570

[27] Henry M Robert III, Daniel H Honemann, Thomas J Balch, Daniel E Seabold, and Shmuel Gerber. 2020. *Robert's rules of order newly revised.* PublicAffairs.

[28] Gregory Rolan. 2017. Agency in the archive: a model for participatory recordkeeping. *Archival Science* 17, 3 (2017), 195–225.

[29] Ansgar Scherp, Carsten Saathoff, Thomas Franz, and Steffen Staab. 2011. Designing core ontologies. *Applied Ontology* 6, 3 (2011), 177–221.

[30] Kristen M. Scott, Simone Ashby, and Adriana Stan. 2020. *Designing a Synthesized Content Feed System for Community Radio.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3419249.3420177

[31] Paul Scriven. 2021. From Tabletop to Screen: Playing Dungeons and Dragons during COVID-19. *Societies* 11, 4 (2021). https://doi.org/10.3390/soc11040125

[32] A Sharma Grover, Osamuyimen Stewart, and David Lubensky. 2009. Designing interactive voice response (IVR) interfaces: localisation for low literacy users. In *The Twelfth IASTED International Conference on Computers and Advanced Technology in Education* (St. Thomas, US Virgin Islands). ACTA Press, Calgary, AB, Canada, 8 pages. http://hdl.handle.net/10204/3882

[33] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering* 25, 1 (1998), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

[34] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. 2012. The NeOn methodology for ontology engineering. In *Ontology engineering in a networked world*. Springer, 9–34.

[35] R. Talhouk, T. Bartindale, K. Montague, S. Mesmar, C. Akik, A. Ghassani, M. Najem, H. Ghattas, P. Olivier, and M. Balaam. 2017. Implications of Synchronous IVR Radio on Syrian Refugee Health and Community Dynamics. In *Proceedings of the 8th International Conference on Communities and Technologies* (Troyes, France) *(C&T '17)*. Association for Computing Machinery, New York, NY, USA, 193–202. https://doi.org/10.1145/3083671.3083690

[36] M Thirumaran, Subham Soni, G Brendha, et al. 2015. Design of context-aware interactive voice response system. *Advances in Natural and Applied Sciences* 9, 6 SE (2015), 669–676.

[37] Thirumaran M. and Banupriya P. 2015. Dynamic Interactive Voice Response System Using Ontology and Java Expert System Shell. *Procedia Computer Science* 70 (2015), 107–113. https://doi.org/10.1016/j.procs.2015.10.049 Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems.

[38] Delvin Varghese, Tom Bartindale, Kyle Montague, Matt Baillie Smith, and Patrick Olivier. 2022. Supporting Real-Time Peer-Mentoring of Rural Volunteers. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 629, 13 pages. https://doi.org/10.1145/3491102.3517598

[39] Aditya Vashistha, Edward Cutrell, Gaetano Borriello, and William Thies. 2015. Sangeet Swara: A Community-Moderated Voice Forum in Rural India. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 417–426. https://doi.org/10.1145/2702123.2702191

[40] Aditya Vashistha, Abhinav Garg, Richard Anderson, and Agha Ali Raza. 2019. Threats, Abuses, Flirting, and Blackmail: Gender Inequity in Social Media Voice Forums. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300302

[41] Deepika Yadav, Prerna Malik, Kirti Dabas, and Pushpendra Singh. 2019. Feedpal: Understanding Opportunities for Chatbots in Breastfeeding Education of Women in India. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 170 (nov 2019), 30 pages. https://doi.org/10.1145/3359272

[42] Deepika Yadav, Pushpendra Singh, Kyle Montague, Vijay Kumar, Deepak Sood, Madeline Balaam, Drishti Sharma, Mona Duggal, Tom Bartindale, Delvin Varghese, and Patrick Olivier. 2017. Sangoshthi: Empowering Community Health Workers through Peer Learning in Rural India. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 499–508. https://doi.org/10.1145/3038912.3052624