Supporting Natural Language Interaction with the Web

Marcos Baez[1234−5678−9012], Cinzia Cappiello[0000−0001−6062−5174], Claudia M. Cutrupi[0000−0001−7933−4721], Maristella Matera[0000−0003−0552−8624], Isabella Possaghi[0000−0002−3782−480X], Emanuele Pucci[0000−0003−2808−5619], Gianluca Spadone[0000−0002−7052−012X], and Antonella Pasquale[0000−0001−7982−2741]

Politecnico di Milano, Piazza L. Da Vinci 32, 20133 - Milano, Italy
[name.surname]@polimi.it

Abstract. Conversational AI is disrupting the way information is accessed. However, there is still a lack of conversational technologies leveraging the Web. This paper introduces an approach to support the notion of Conversational Web Browsing. It illustrates design patterns for navigating websites through conversation and shows how such patterns are sustained by a Web architecture that integrates NLP technologies.

Keywords: Conversational AI · Conversational Web · Conversational Design Patterns.

1 Introduction

Conversational agents (CAs) are pervading a broad range of activities, as their natural language (NL) paradigm simplifies the interaction with digital systems. They offer benefits in different situations where users may take advantage of voice-based interaction for accomplishing their tasks [8, 3]. Recent works are capitalising on this technology, for example to design voice-based CAs for searching the Web [2], to automatically generate CAs out of a website content [9], or to enable end users to customize their CAs for the Web [5]. This interest shows the feasibility and potential of NL interaction for making the Web truly for everyone. However, very often the CA development and deployment is detached from Web architectures: CAs are seen as tools that complement the Web access experience by providing additional content, not granting access to the website content itself. There can be situations, instead, where some forms of disabilities (whether permanent, temporary, situational) demand for a voice-based access to the website. Despite this need, there is still a lack of proposals for a full-fledged integration of Conversational AI within Web architectures.

This paper tries to fill this gap by illustrating the conceptual architecture of *ConWeb*, a framework for *Conversational Web Browsing*. The idea is to enable users to navigate content and services accessible on the Web by "talking to websites" instead of browsing them visually, by expressing their goals in natural language and accessing the websites through a dialog mediated by a conversational agent (e.g., a voice-based browser plugin). ConWeb tries to respond to

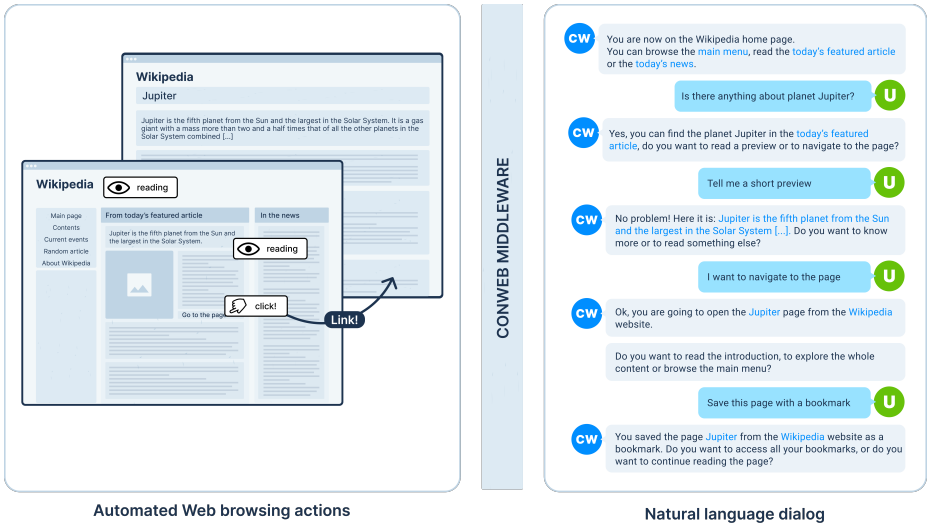

Fig. 1. An example of a Conversational Web Browsing on Wikipedia. The reported conversation recalls the conversational patterns resulting from the user research [1].

some requirements that we identified through an extensive human-centred process. With the help of 26 blind and visually impaired (BVI) users, along different sessions of interviews and co-design workshops, we were able to identify and validate some prominent challenges and some related interaction-design patterns sustaining the notion of Conversational Web Browsing [1].

After illustrating the notion of Conversational Web Browsing and highlighting the main challenges behind the provision of this new paradigm (Section 2), in Section 3 this paper outlines the main results of the conducted user research. Section 4 illustrates the design of *ConWeb*, a software platform for the Conversational Web that integrates information models, NLP models and other components of a Web architecture to manage a voice-based dialogue organized around the identified patterns for conversational Web browsing. Section 5 finally draws our conclusions and outlines our future work.

2 Conversational Web Browsing

To explain the main idea behind Conversational Web Browsing, we illustrate a scenario of a user browsing the Wikipedia Home Page[1] by dialoguing with a conversational agent (e.g., a smart speaker or a voice-based browser plugin). As represented in Fig.1, starting from the home page the user can be introduced with a short description along with the main organisation of the website. The user could also at any point get oriented by inquiring about the content available in a given context, e.g., by uttering *"Is there anything about planet Jupiter?"*.

[1] https://en.wikipedia.org/wiki/Main_Page

The user can then *navigate* the website by following up on one of the available options (e.g., *"I want to navigate to [...]"*). These requests can trigger navigation within or across pages in the website (e.g., from the Home to an article page). Ultimately, the user can browse the structure of the content or directly read the available content. To enable such interaction, a middleware sitting between the user and the website should be able to identify the offerings and content of the website that can be accessed through the conversational medium, interpret user *intents* and associated *entities* from user utterances, and automatically perform related *actions* on the website (e.g., click, extract information).

This paradigm is one of the few emerging approaches exploring the integration of conversational capabilities into the Web [9, 2, 5]. Previous work explored basic issues posed by a tight integration of Conversational AI with the Web, related to automating Web browsing actions to respond to NL user commands [4], with a focus on technical feasibility. This paper tries to give a further contribution by discussing how to support more articulated *design patterns* for conversational Web browsing that were identified through an extensive user research [1]. Incorporating conversational patterns is fundamental to support recurrent sequences of human-bot interactions [7] serving expected browsing tasks. The following sections illustrate the user requirements and the technical aspects that pattern integration implies.

3 User Requirements

In the time period from April to September 2021 we conducted a user research that involved 26 BVI users by means of online structured interviews and in-presence focus groups and co-design sessions. We first asked them to describe their experience with current voice-based assistive technologies. By using online tools for CA rapid prototyping (e.g., DialogFlow), we then solicited them to express their desiderata on the design of novel CAs for accessing websites. A complete description of the study is reported in [1]. In the following we illustrate some of the identified conversation *patterns*, which mainly refer to the structure of conversation for Web browsing. These patterns guided us in extending a preliminary version of the ConWeb prototype [4] to: *i)* support an incremental, dialog-based exploration of the Website, and *ii)* grant flexibility in the dialog organization, to fulfil the need of personalized browsing experiences.

Shaping-up the map of the navigable space. Users claimed that learning the structure of the website is a crucial initial step when they access a website for the first time. For this reason, they asked for strategies to identify *high-level navigation mechanisms* to support them in understanding the website structure, and fluidly move along the main areas (e.g., in Figure 1: "You can browse the main menu, [...]"). To identify how to move along different information nodes, they highlighted mechanisms for *link predictability* (e.g.: "Do you want to read a preview, or [...]") and for *keeping track of the navigational context* (e.g.: "You are now in the Wikipedia Home Page.").

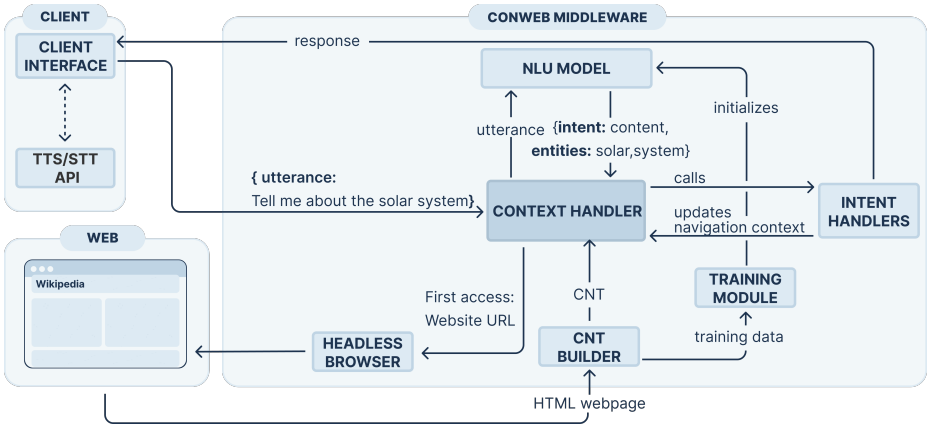

Fig. 2. Conceptual architecture of the ConWeb platform.

Navigating through intelligible and quick mechanisms. Depending on their tasks and preferences, participants demanded for different navigation strategies. They described *in-depth explorations* to narrow down navigation options along the hierarchy of nodes, but especially *punctual, fast-served requests* were discussed as a means to locate a desired content (e.g.: "Is there anything about [...]"), along with the capability of *bookmarking information nodes* (e.g.: "Save this page as a bookmark") for a direct access to content of interest.

Summarizing and segmenting the page content. The research conversational paradigms should *prevent unwanted and unneeded explorations* resulting in poor user experiences. *Segmenting contents and highlighting characterizing keywords* could help localize the content of interest (e.g.:"Jupiter is [<*short content preview*>]. Do you want to know more or reading something else?").

Providing access to conversation-scaffolding intents. Users frequently expressed the need for *scaffolding intents* to help them identify possible actions at different navigation levels and for the provision of *feedback on the system status*, such as the use of landmark cues.

4 The ConWeb Platform

Translating the identified interaction patterns into architectural choices for the design of the ConWeb platform has required focusing on the following aspects:

- A conversational-browsing model must be built when the website is first accessed, to index and present to the users the available conversation nodes and the navigation structures that can sustain conversational browsing.
- A conversation node does not necessarily correspond to an entire Web page; it can be a content paragraph, a navigation menu, a link, or any other element in the Web page that can be presented independently from the others and has a role in the progressive exploration of the website content.

- A context representation characterising the navigation status must be handled to let the users to move easily backward, i.e., along previous conversation nodes, and forward, i.e., to identify and explore new reachable nodes.
- To extract browsing-relevant intents and entities from the user utterances, an NLP engine must be adequately trained starting from the website content.
- Recognized intents and entities must be matched with navigation- and content-reading actions as deriving from the conversational-browsing model.
- The resulting CA must recognise website-specific intents as well as scaffolding intents related to auxiliary commands for the user to control the conversation.

Figure 2 illustrates the resulting conceptual architecture. The current version of the framework has been implemented by embedding the conversational capabilities into a traditional browser. In particular, it serves its logic via a client that uses external APIs (Google APIs in the current prototype version) for Text-To-Speech and Speech-To-Text translation. It gathers the NL user utterance and builds a request that embeds additional user session parameters needed to control and handle the user navigation context. At the server-side, an NLP engine (RASA in the current implementation) parses the user utterance to extract browsing intents and entities. Also considering the navigational context, the framework identifies and performs the necessary navigation actions. For example, for the user's request "Tell me about the solar system", the intent is a `content request`, and `solar` and `system` are the entities extracted. It is thus possible to identify the conversation node where those entities can be found, and to navigate automatically towards that node. The automatic navigation is performed through a headless browser (Selenium in the current prototype), which starts a browser session every time a new page is accessed. As a fundamental step to manage navigation, a context handler parses the page HTML to create a model that can help map the interpreted intents and entities onto contextual navigation actions. By invoking proper intent handlers, this module also builds the responses that the client renders in form of conversation. This mapping is the most challenging aspect. As described in the following, it requires integrating the NLP pipeline with the execution of browsing actions contextualised for the accessed website.

4.1 Conversation-oriented Navigation Tree (CNT)

When a website is accessed for the first time, a conversational-browsing model is built to index the available content segments and the navigational structures that can sustain conversational browsing. This is needed for providing the user with an overview of the available content, for allowing her to move within the available content space based on the tracked navigational context, and for enacting a direct navigation towards a content responding to a search key.

As reported in Figure 3, for each accessed Web page a Conversation-oriented Navigation Tree (CNT) is built to represent the hierarchical nesting of different page elements, the *conversation nodes*. Conversation nodes represent content

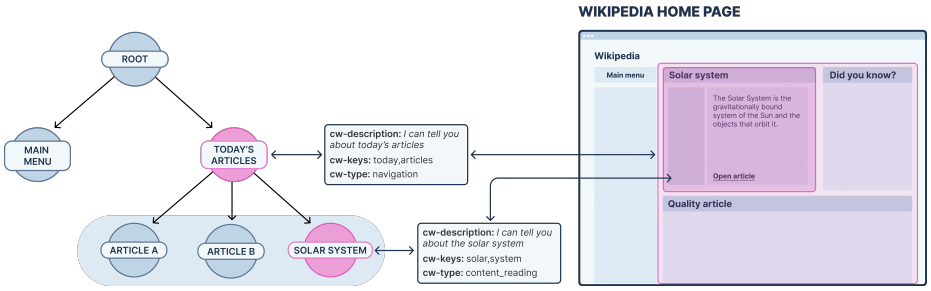

Fig. 3. Conversation-oriented Navigation Tree

segments (e.g., the leaf nodes in Figure 3, representing the available articles), or navigational indices providing access to content segments (e.g., the *Today's articles* index). Their granularity serves the purpose of building a dialog for the incremental exploration of the website content. Each CNT node also specifies attributes and descriptions extracted from the Web page, which can help render the node content through conversation. The CNT traversal then supports both the progressive visit or the direct access (i.e., by means of search keys) to the specified conversation nodes. Pertinent conversation nodes are identified by matching the intent and entities extracted from the user's utterances with the descriptions and keys summarizing the content of a node.

4.2 Building the CNT

The specific content and the descriptive attributes stored in each CNT node are extracted from the HTML code of the accessed Web page. Based on the user request, a headless browser simulates navigation actions and downloads the HTML code of a page. The extraction of many of the CNT elements can be performed automatically, by parsing the HTML code. However, to make the process more effective, specific annotations can augment the page HTML to tag and specify the elements useful for building the tree. For example, content-oriented tags can be used to add short summaries of content segments (`<cw-description>`), and representative keys (`<cw-keys>`). A type tag (`<cw-type>`) then characterizes page segments as *content* or *navigation* elements, with the latter providing navigational structures for indexing content nodes. These annotations can guide the page interpretation process and the extraction of relevant CNT elements.

4.3 Training the NLP model

The NLP model for intent classification and entity extraction has to be trained for each website. While the recognition of some intents and entities (e.g., those for scaffolding commands) is content-agnostic and can be handled in the same way for any Web site, i.e., it does not need specific training, for some others (e.g., content-access requests) the NLP outcome depends on the knowledge on

the specific Web site content. For this reason, in addition to representing the hierarchy of conversation nodes, the CNT also specifies the "domain knowledge" that can be used for the automatic generation of a website-specific training data set. Starting from the keys indexing each conversation node, a list of training sentences for each content-access intent is automatically generated and used to train the NLU model for intent classification and entity extraction.

4.4 Handling the classified intents

To perform proper navigation actions and build conversational responses to the user's requests, ConWeb includes a library of *Intent Handlers* that are invoked depending on the outcome of the intent classification and entity extraction. For example, a *Navigation handler* enables moving along the entire CNT to localize a conversation node when a content-access intent is recognized, whereas a *Link handler* and a *Content Reading handler* build the conversation for presenting nodes of type navigation and content, respectively: if the user request is for accessing a content node, after localizing the node in the tree the Content Reading handler build the dialog for rendering the node description; if instead the request refers to a link traversal, then an automatic navigation to the corresponding page is enacted together with the provision of feedback messages to inform the user of the content of the target node. A *Scaffolding Intents* handler serves commands available at any conversation node, such as those for getting information on the current page, help, back and forth commands, access to the bookmark list.

Intent handlers enforce separation of concerns and grant flexibility: they make it easy introducing additional conversational patterns, for example to manage Web page components not yet covered by our current prototype (e.g., forms or image-reading intents). Other extensions of the intent-handler library can also be conceived to accommodate specific users preferences, for example related to varying text-reading styles. This last feature responds to the need for personalization recurrently remarked during the conducted user research [1].

5 Conclusion

This paper has discussed interaction and architectural patterns that can make the Web accessible through an NL interaction. It has shown how Conversational AI can be integrated within Web architectures, to provide an additional channel for accessing websites. So far, our work has mainly focused on content-oriented websites. Even if further elements are needed to cover the requirements posed by other website categories, we are confident the flexibility offered by the ConWeb architecture can favour the extensions needed for handling other types of intents. There are however some limits that will be investigated in our future work.

The performance for training on the fly the NLU model has to be improved. When a website is accessed for the first time, the current prototype requires up to 10s to generate the training data set and update the model. Our future work will focus on identifying lightweight techniques to reduce the response time.

The current ConWeb prototype requires augmenting the website HTML with the CNT annotations. We are now designing proper authoring environments and also want to capitalize on standardization activities[2] that are proposing HTML extensions for accessibility. For granting conversational access to any website, even those not properly augmented, we are working on deriving CNT elements through the automatic extraction/summarization of the website content.

Our future work will be devoted to user studies that will also address sighted users, to understand if the assumptions derived with and for BVI people can be extended to other classes of users. Involving BVI users has allowed us to identify the most stringent requirements for a conversational browsing detached from the visual channel. However, in line with recent initiatives [6], we are confident the resulting approach can benefit people universally, and has a potential that will impact Web Engineering in the coming years.

Acknowledgments

We are grateful to the associations UICI, ADV, Real Eyes Sport for their help in the user research. This paper is dedicated to Prof. Florian Daniel, who suddenly passed away in April 2020. He first had identified the value of this research.

References

1. Baez, M., Cutrupi, C.M., Matera, M., Possaghi, I., Pucci, E., Spadone, G., Cappiello, C., Pasquale, A.: Exploring Challenges for Conversational Web Browsing with Blind and Visually Impaired Users. In: CHI'22 Extended Abstracts. ACM (2022)
2. Cambre, J., et al.: Firefox Voice: An Open and Extensible Voice Assistant Built Upon the Web. In: CHI 2021. pp. 1–18 (2021)
3. Chang, Y., et al.: Tourgether: Exploring Tourists' Real-time Sharing of Experiences as a Means of Encouraging Point-of-Interest Exploration. ACM Interact. Mob. Wearable Ubiquitous Technol. **3**(4), 128:1–128:25 (2019)
4. Chittò, P., Baez, M., Daniel, F., Benatallah, B.: Automatic generation of chatbots for conversational web browsing. In: Proc. of ER'20. pp. 239–249. Springer (2020)
5. Fischer, M.H., Campagna, G., Choi, E., Lam, M.S.: DIY assistant: a multi-modal end-user programmable virtual assistant. In: PLDI '21. pp. 312–327. ACM (2021)
6. Microsoft: Inclusive design (2022), https://www.microsoft.com/design/inclusive/
7. Moore, R.J., Arar, R.: Conversational UX design: A practitioner's guide to the natural conversation framework. Morgan & Claypool (2019)
8. Pradhan, A., Mehta, K., Findlater, L.: Accessibility Came by Accident. Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In: CHI 2018. pp. 1–13 (2018)
9. Ripa, G., Torre, M., Firmenich, S., Rossi, G.: End-user development of voice user interfaces based on web content. In: IS-EUD 2019. pp. 34–50. Springer (2019)

[2] See https://schema.org/SpeakableSpecification