Pre-print of full paper presented at CONVERSATIONS 2021 – the 5th International Workshop on Chatbot Research, an online virtual event hosted by the University of Oslo, Norway, November 23-24. The final version of the paper will be published in the forthcoming Springer LNCS post-workshop proceedings.

Socially Aware Interactions: From Dialogue Trees to Natural Language Dialogue Systems

Maria Inês Lobo[1], Diogo Rato[1], Rui Prada[1], and Frank Dignum[2]

[1] INESC-ID and Instituto Superior Técnico, University of Lisbon, Portugal
[2] Umeå University, Sweden

Abstract. In this paper, we present a prototype of a human-agent dialogue system, in which the scenarios are easy-to-author, as in tree-based dialogue tools. These, however, only allow for scripted and restricted dialogues. For this reason, we focused on developing a flexible and robust deliberation mechanism as well, based on the Cognitive Social Frames model and the theory of social practices, so that the conversational agent could provide acceptable responses according to different social contexts. Having access to sequences of frames containing small dialogue trees, the agent activates the most salient frame to reply appropriately to the user's input. As a proof of concept, we designed a medical diagnosis scenario between a doctor and a patient in which the agent could play both roles given different settings of the scenario. In this prototype, the user had to choose from a limited set of alternatives, based on the current context, in order to respond to the agent; however, in the future, we intend to allow users to write freely, expecting to be able to map their utterances to the appropriate context.

Keywords: Conversational Agent · Context-Aware Dialogue · Social Context · Dialogue Authoring · Dialogue Training · Medical Diagnosis

1 Introduction

There is a growing demand for realistic social behaviour of agent systems in a variety of computational fields [7]. In this document, we will focus on dialogue systems. These systems should be aware of their social context in order to cooperate effectively with humans and other AI systems, meaning that social aspects of the context should be considered to manage interactions between these systems and other entities. This will allow for an efficient and focused dialogue aimed towards a specific goal that is endorsed by all participants of a given interaction.

In tree-based dialogue tools, the social context is taken into consideration solely through the authors of the dialogue, meaning that the conversation lacks flexibility. For instance, the Dialogue Trainer software[3] is a system that allows for authoring very simple yet directed dialogues, with limited choices available,

[3] https://www.dialoguetrainer.com/en

to train medical students throughout the process of an anamnesis. In this tool, the responses of the conversational agent are scripted by an author in a large dialogue tree, so the agent does not need to adapt to changes in the context, displaying a restrictive and unrealistic behaviour [17]. Natural language dialogue systems, on the other hand, are flexible and capable of managing user errors, but their authoring mechanisms are typically not intuitive or general enough, preventing authors from easily configuring dialogue data for different settings. With this in mind, we would like to address the following research question:

> Is it feasible to create a flexible and robust dialogue system that adapts to changes in the social context, while maintaining the authoring property of tree-based dialogue tools?

Our main goal will be to find a middle ground between the identified dialogue systems or tools, in order to model and develop a system that meets the requirements of flexibility, robustness and authoring. Then, we will be required to use the social context to manage a dialogue between a user and an artificial agent, based on defined social practices and identities, norms, conventions and values. This will enable us to build flexible and robust dialogue management systems that take the human as a starting point and that are necessary to create a human-based user interface. Moreover, we will have to ensure that the authoring property of the system is maintained, since it is essential for the quick set up of the domain knowledge and resources required to support the agent.

Considering the prior statements, we designed and implemented a dialogue system's software prototype based on the Cognitive Social Frames model [12], as well as the theory of social practices [13]. The key contribution of our work was how we decided to organize the data in the system to enable authorable socially-aware dialogues. We attempted to make it intuitive, controllable and general, so that dialogue scenarios could be easy-to-author and, simultaneously, be given as input to a conversational agent that deliberates and exhibits an appropriate behaviour according to different social contexts. These could vary based on location, time, conversation state, type of relation, and many other factors. For example, it could be as simple as saying "good morning" or "good afternoon" depending on the time of the day, or, in an anamnesis setting, being more formal to an unfamiliar doctor as opposed to being more friendly to a doctor the agent already knows and trusts. Additionally, we selected Twine[4], a widely known authoring tool, to configure the dialogue scenarios of the system. Finally, we defined a medical diagnosis scenario to demonstrate how the system operated.

2 Background

The human-agent interaction field, where agents should operate as partners to humans, as well as serious games and social simulations, all require more believable social behaviour of agent systems [7].

[4] https://twinery.org/

Dignum et al. [7] proposed five key ingredients to generate realistic social behaviour in agents: social motives, identity, skills, values and social reality. In essence, social agents should have personal goals and preferences, while abiding by the norms and values of the social group to which they belong. Moreover, they should have their own knowledge and beliefs about the world, as well as the ability to recognize and understand other agents. Given these components, the main idea is to design believable social agents that have sociality at the centre of their reasoning, rather than simply adding "a few social modules" to their architectures.

We will start this section by outlining the social context and its dimensions. Following that, we will introduce social theories based on social identities and practices, which support the development of agents that adapt to the social context. Additionally, we will go over a few models and architectures inspired by these theories, culminating in a section on the Cognitive Social Frames model.

2.1 Social Context

Dey reviewed former attempts of defining context within the field of computer science, finding that previous definitions were extremely challenging to apply in the context-aware computing area [5]. As a result, Dey proposed a generally accepted definition of context, aiming to help software developers easily outline the context for a specific application scenario:

> "Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

Zimmermann et al. found the prior definition to be too general, deciding to enclose this same concept "by a formal and an operational part" [20]. In regards to its formal extension, they identified five fundamental categories to classify the context information:

- **Individuality**: Includes the characteristics of the entity to which the context is associated.
- **Activity**: Entails the entity's current and future goals, tasks and actions. Tasks are goal-oriented and can comprise operation sequences with concrete goals.
- **Location**: Describes an entity's physical or virtual location, and other spatial details, such as speed and orientation.
- **Time**: Describes an entity's time information, like the user's time zone, the current time, any virtual time, time intervals, and so on.
- **Relations**: Depicts the relations of an entity with other entities. Depending on the relation, each entity plays a particular role.

In terms of the context's operational extension, Zimmermann et al. highlighted its dynamic properties, caused by context transitions (e.g., changes in the relevance or accessibility of context attributes), or by sharing contexts between multiple entities (e.g., establishment of new relations or knowledge exchange). Context is then defined using a structured approach, with the goal of "bridging the user-developer gap" – giving users an intuitive understanding of this concept and facilitating developers' implementation of a context model for context-aware systems.

Based on this understanding of context, Zimmermann et al. highlighted a few context-aware applications in [21]. These included a museum guide system that provided personalised audio information to users regarding the exhibitions, based on their location, head rotation, interest, among other factors, as well as an advertising board system that changed its content based on the noise level, time of day, or whether or not people were paying attention. These contextualized and personalized systems adapted not only to the current environmental state, but also to the user's needs and preferences, being able to support and relate to the humans who surrounded or used them. Nevertheless, we can assume that the described context categories are suitable to describe the agent of a system as well.

There is also an ongoing concern about establishing trustworthy relationships between agents and users in these types of systems. Zhao et al. [18,19], for example, developed a rapport management model, tying conversational strategies, associated with verbal and non-verbal cues, to specific goals that, when achieved, resulted in rapport. This model was eventually incorporated into a socially-aware virtual assistant.

2.2 Social Identity Theory and Social Practice Theory

Social Identity Theory (SIT), formulated by Tajfel and Turner [14], relies on "the group in the individual" [15]. It proposes that people exhibit all sorts of "group behaviour", such as cooperation, within their in-group and prejudice against out-groups. Individuals identify with groups or social categories for self-reference, following the prototypical behaviour of the group to which they belong [9]. As a result, the group's social status and prestige will influence the self-esteem of its members. Therefore, as an attempt to increase the reputation of their in-group, they will engage in a "process of social comparison with relevant out-groups".

Self-Categorization Theory also shares the concept of social identity, however it proposes that personal and social identities represent distinct levels of self-categorization, depending, respectively, on whether a person is in the presence of only their in-group or if more social categories are available [15]. Then, when a shared social identity becomes more salient, people start perceiving themselves as representatives of a shared social category membership rather than distinct individuals with their own personality [16].

These two theories have inspired various models for designing and developing social agents, such as the Dynamic Identity Model for Agents [8] or the Cognitive Social Frames framework [12], explained in depth in Section 2.3. In addition, Dignum et al. developed an abstract architecture for social reasoning [6], using not only the Social Identity Theory, but also the Social Practice Theory.

Practice Theory is a type of cultural theory, in which social order is rooted in collective cognitive structures - a “shared knowledge” - allowing to ascribe meaning to the environment in a socially shared manner [13]. Its “smallest unit of social analysis” is the practice, a “routinized type of behaviour” constituted by various interlinked and mutually dependent elements: physical and mental activities, resources and their use, and background knowledge required to understand the world’s objects and agents.

Holtz [10] applied this theory in an agent-based model aimed at generating social practices to understand resource consumption behaviours. In this approach, social practices were viewed as independent social structures useful to provide valuable insights into a range of social behaviours, however they can also be used as “input and filter on the individual’s deliberation process” [6], as in Dignum et al.’s previously mentioned architecture.

2.3 Cognitive Social Frames

Cognitive Social Frames (CSFs) is defined by Rato et al. [12] as “a framework that enables the adjustment of the agent’s cognition” depending on the social context. The agent’s cognition is enclosed into abstract blocks known as Cognitive Resources, which can hold knowledge and processes used to guide the agent’s behaviour.

According to the model, the agent has to follow specific steps. The agent starts by observing the environment with its sensors, resulting in a set of perceptions that are sent to the agent’s Sensory Memory. Then, the agent filters the previous perceptions given the salient CSF, transforming them into social perceptions and constructing the Social Context, which is transferred to the agent’s Working Memory. Following this, the agent updates the previous salient CSF, selecting the CSF with the highest salience – computed as a balance between fitness to the social context and personal preferences of the agent – from the set of CSFs in the agent’s Long-Term Memory. So, the salient CSF in the Working Memory is updated, meaning that the set of deployed Cognitive Resources, associated with the salient CSF, is also modified. Finally, the updated collection of deployed Cognitive Resources is executed, and the agent will now reason, decide or interact with the environment in accordance with the social context.

3 Related Work

Dialogue training tools are widely used in a variety of fields, existing a particular interest in tools to train medical students, as we will see along the following paragraphs. In this section, we will start by discussing the benefits and drawbacks of using tree-based dialogue training tools. Then, we will describe natural language dialogue systems that seek to address the problems associated with scripted dialogues. Lastly, we will highlight ongoing concerns about these systems.

3.1 Scripted Dialogues

Communicate! [11] is a serious game for developing social skills, in particular for training health-care professionals – players – in their interactions with patients – agents – with the goal of improving the communication between them. These entities will be required to speak in a scripted dialogue, such as a tree-like structure, and players will be given options for each utterance. Furthermore, when a scenario ends, feedback is given. Agents' emotions e.g., facial expressions, provide immediate feedback as well. In addition, scenarios can be authored, meaning that utterances, feedback, emotional reactions, scores and other parameters can be configured in an editor. With that said, the Dialogue Trainer Software, mentioned earlier, shares the same attributes of the prior game, being a commercial tool for communication training.

Nevertheless, as identified by Weideveld, there are a number of problems in using the previous tools to teach adequate communication skills [17]. From the user's standpoint, since the dialogue is not self-made, players will be unable to properly apprehend relevant communication abilities such as clarity, understanding and appropriateness. Furthermore, in the agent's perspective, given that their responses are scripted, the dialogue ends up being predetermined, implying that the agent does not need to adapt to changes in the context, or, better yet, the "adaptive" behaviour of the agent is controlled by the editor of the scenario, which results in a lack of authenticity and flexibility. Finally, from the viewpoint of the editor, while the dialogue content is easy to author, managing a big dialogue tree may be challenging and repetition is prone to happen – the same utterances may have to be specified more than once in distinct conversation paths.

3.2 Context-Aware Conversational Agents

Inspired by the Communicate! tool, the Social Agents for Learning in Virtual Environments (SALVE) [1] game was developed. SALVE is a dialogue-based serious game in which players can experiment various scenarios and act in different roles, while communicating with an agent, whose avatar is shown in a graphical interface. The agent expresses emotions and reactions in response to the user's inputs, detecting also repetition during the conversation. In addition, the game allows to evaluate the player's abilities, scoring their empathy, social interpersonal behaviour, and communication skills, among others.

In SALVE, the agent interprets the players' utterances using AIML and a rule-based approach, together with the concept of Social Practices, as a way of restricting the scope of the dialogue [17]. Then, the game scenarios will follow predefined social practices (e.g., the practice associated to an anamnesis scenario, as seen in Table 1 [2]), and the agent will be able to appropriately respond to the user, due to its pattern matching mechanisms.

In relation to the previous scenario, a dialogue system including a virtual patient was recently developed by Campillos-Llanos et al. [4], using a "knowledge-based and rule-frame-based approach". A medical instructor could edit the patient's medical record in a simple interface, allowing the system to run multiple

Table 1. Example of a Social Practice in an anamnesis scenario [2]

Abstract Social Practice	**Doctor Patient Dialogue**
Physical Context	
Resources	current time, medical instruments
Places	hospital, office
Actors	user, agent
Social Context	
Social Interpretation	consulting room, consulting time
Roles	doctor, patient
Norms	patient is cooperative (gives truthful and complete answers), doctor is polite
Activities	welcome, presentation, data gathering, symptom description, speech acts
Plan patterns	Welcome, Presentation, Data Gathering, Symptom Description, Therapy
Meaning	support the patient, create trust, eliciting patient's problems and concerns, empathic response
Competences	listening effectively, being empathic, use effective explanatory skills, adapt conversation

medical cases by modifying the agent's health state (e.g, symptoms, lifestyle). The user of this system, commonly a medical student, could type text to the agent, which would consequently be processed by a natural language understanding module. The dialogue manager would then receive a semantic frame containing information about the dialogue state and context information of the user's input, querying this to find the appropriate answer. According to this answer, the agent selected relevant information from the patient's record, replying to the user using a template-based generator.

These systems are clearly an improvement compared to scripted dialogues: not only do they allow users to enter their own input with the keyboard, but they are also more flexible and capable of handling errors in the conversation. Authoring new or existing scenarios, on the other hand, becomes more difficult, especially if no authoring tool is available, as in SALVE. Regarding the second system, while it is possible to author different health states for the agent, the system is still limited to the domain of a medical diagnosis. As a result, no new scenarios can be created; instead, only the content of an existing one can be altered. Given these examples, what is lacking is an efficient method of structuring dialogue data so that it can be easily configured and controlled by its authors in a variety of settings, while also being accepted as input by the deliberation process of a socially-aware agent. Then, we will need to search for a compromise between scripted and rigid dialogue tools - where authoring is usually simple, enabling for different scenarios to be configured - and natural language dialogue systems - where flexibility and error handling are present, however intuitive authoring methods that can be used across different domains are absent.

4 Dialogue System

We considered flexibility, robustness, and authoring to be the most important requirements for modelling and implementing this prototype of a dialogue system. Tree-based dialogue tools restrict the flow of the conversation by preventing it from taking various alternative paths. A flexible dialogue system, on the other hand, will adapt to changes in the conversation's context, allowing it to be less rigid and, consequently, more natural to the user. This adaptation mechanism should be robust, meaning that the system should be able to generalize to different contexts without collapsing. Robustness also includes the system's ability to handle user errors or unexpected actions. Moreover, it should be possible to create new dialogue scenarios or edit existing ones in a simple and controllable manner, given that the easier it is to author, the more people will be able to use the system, not being required to have any particular technical skill to do so.

An additional requisite we considered was for users to be able to easily play different roles in a conversation, allowing them to experience the same scenarios from multiple perspectives.

We applied the Cognitive Social Frames model, combined with Social Practices, in order to achieve these goals. Regarding the Social Context components, we followed Zimmerman's definition of context and its five fundamental categories. The following sections detail the agent's deliberation cycle, as well as our implementation and authoring process. In addition, we specified an anamnesis scenario to demonstrate an application example of the developed tool.

4.1 Deliberation Cycle

The agent's architecture is depicted in Figure 1 and it was greatly inspired by the Cognitive Social Frames model. The key elements of the agent's deliberation cycle are the social practices, the frames, the cognitive resources, the context and the knowledge base.

Both the agent and the user have access to a "shared knowledge", in the form of social practices, conventions and values, aiming to limit the scope of their interaction. In a dialogue, social practices include collections of interlinked frames, whose connections convey expectations on how the conversation should follow, just like the "plan patterns" in Table 1. Then, for a particular scenario, we assume that one or more social practices are available. Concerning the frames, they are composed by a set of cognitive resources, which represent dialogue pieces – small dialogue trees with one or more utterances – created given the norms, motivations and beliefs of the dialogue's entities.

Regarding the context, as mentioned before, we considered Zimmerman's five dimensions of context. Then, in a human-agent dialogue, we propose that the context presents itself like this:

- **Time**: time of the day (e.g., morning, night)
- **Location**: physical location (e.g., university, hospital)

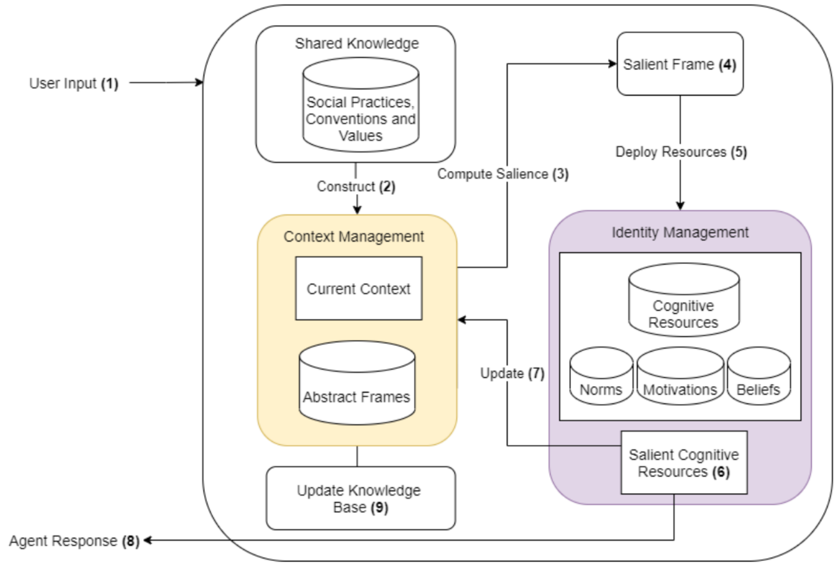

Fig. 1. Agent's architecture in the dialogue system

- **Activity**: state of the conversation (e.g., greeting, farewell), as well as goals, which, in this case, are associated with specific information that the entities of the conversation want to acquire (e.g., name, age, particular symptoms if in a medical scenario)
- **Relation**: social roles (e.g., doctor/patient, teacher/student), and type of relation (e.g., formal, friendly)
- **Individuality**: emotional state (e.g., being happy, being angry) – physical features were not yet considered given that we are only using written dialogue

These components of the context will work as a "filter", which means that, by mapping a given input to a context, the agent will be able to filter the most relevant frames that are appropriate for that same context, providing an acceptable response. As for the knowledge base, it simply represents the information (e.g., name, age, symptoms) that the agent manages to obtain throughout the conversation.

As shown in Figure 1, the user starts by sending their input to the agent **(1)**. After receiving it, the agent is able to translate this input into the relevant dimensions of the context **(2)**, updating it accordingly – Context Management. Having access to sets of interconnected frames and considering the updated context, the agent computes the salience of each frame **(3)**. The most salient frame **(4)** is then selected and its cognitive resources are deployed **(5)**. The agent proceeds to identify the salient cognitive resource **(6)** from the collection of deployed resources, so that an appropriate response can be given – Identity Management. Finally, given this resource, the agent responds to the user **(8)**. In addition, the context will be updated again **(7)** based on the salient frame. The knowledge base may be modified throughout this cycle as well **(9)**, depending on whether relevant information is provided by the user or agent in their dialogue.

4.2 Application

The dialogue system application, implemented in Python, comprises the dialogue interface and deliberation cycle of the agent. In the dialogue interface, we list the sentences - associated with the cognitive resources - that the user can select at a specific step of the conversation, given the current context. In its deliberation process (Algorithm 1), the agent will receive the user's input and deliver an appropriate response, meeting the system's flexibility requirement.

The agent begins by updating the current context and the knowledge base, repeating this update close to the end of the algorithm as well. If the agent detects that the current dialogue tree has not been visited all the way through, it will continue on the current tree, following its course. Otherwise, it will calculate the salience of the frames in order to determine the resources that the agent could use as a response. The salience of a frame is calculated by the following formula:

$$Salience_{frame} = Frequency_{frame} + KnowledgeBaseMatch_{frame} \quad (1)$$

The frame frequency corresponds to the number of times the frame appears after the current frame, which is the one associated with the current context. The resulting set of salient frames will have to be filtered to select the ones whose resources can be used by the agent, according to its role. If no frames are found, the agent will stay in the current frame. Therefore, the agent may be able to move through the practices, by altering the frame, or simply follow the lead of the user, by remaining in the same frame. In addition, if more than one frame is salient, the agent will randomly select the most salient frame, the same applying to the resources' selection. In the end, the agent returns the sentence associated with the most salient resource.

Following this cycle, the agent's response appears in the dialogue interface, and the user's possible inputs are updated based on the agent's response, using a reasoning process similar to the agent. There is no need to randomize the frames and cognitive resources in this case, since the user should have all the possible options, given the current context, available.

Besides being flexible, the dialogue system also demonstrates robustness, by being able to handle different contexts and dialogue scenarios without breaking, due to its general and simple deliberation mechanism. We also took into consideration the possibility of user error, which is quite common in dialogue systems. Thus, the system should include an error management mechanism that can perform error detection and recovery, i.e., the ability to recognize and respond to mistakes.

At the moment, we only implemented the "timeout error": if the users take an unusually long time to answer, the agent should acknowledge that it is waiting for their response. Having access to a timeout frame that is not linked to other frames, the agent can activate the respective frame at any time during the

Algorithm 1 Agent's Deliberation Cycle

```
function AgentDeliberation(UserInput)                                          (1)
    CurrentSocialContext = UserInput.Tags                                       (2)
    KnowledgeBase.Add(UserInput.Knowledge)                                      (9)
    PossibleResources = UserInput.NextResources
    if Length(PossibleResources) = 0 then
        CurrentFrame = FindFrame(CurrentSocialContext)
        SalientFrames = ComputeSalientFrames(CurrentFrame)                      (3)
        PossibleFrames = CheckFramesRole(SalientFrames, AgentRole)
        if Length(PossibleFrames) = 0 then
            PossibleFrames.Add(CurrentFrame)
        end if
        SalientFrame = Random(PossibleFrames)                                   (4)
        CurrentSocialContext = SalientFrame.Tags                                (7)
        PossibleResources = CheckResourcesRole(SalientFrame.Resources,
AgentRole)                                                                      (5)
    end if
    SalientResource = Random(PossibleResources)                                 (6)
    KnowledgeBase.Add(SalientResource.Knowledge)                                (9)
    return SalientResource.Sentence                                             (8)
end function
```

The right-aligned comments correspond to the numbered steps in the agent's architecture (Figure 1)

conversation. This enables the agent to manage errors in the conversation in a robust and flexible manner. Then, when it is the user's turn in the conversation, the agent begins counting the time until the user responds. If the user does not say anything within the timeout value defined in the timeout frame (error detection), the agent will enter this frame and select one of its cognitive resources' utterances to say, checking if the user is still present (error recovery). The user's options will be updated based on the agent's utterance. Finally, once the user replies, the agent will repeat what it said before entering the timeout frame, restarting the conversation from where it was left (error recovery).

This same mechanism of having one general, unconnected frame, that deals with a single unexpected event, could be applied to other errors in the user's discourse, such as repetitions, contradictions, providing insufficient or excessive information in their utterances or switching to topics unrelated to the current context [3].

4.3 Authoring

The data that the agent requires to manage a dialogue with a user is organized in the system's architecture in such a manner that configuring it in an authoring tool is fairly straightforward, thereby satisfying the authoring requirement mentioned earlier. For this purpose, we selected Twine as an authoring tool, given that it is an easy-to-edit storytelling platform that enables us to represent all of the architecture's elements. A Twine add-on allowed us to export the data from the editor to a JSON file, which was then imported and read by the system's application, in order to store the frames, cognitive resources, and social roles.

Twine allows editors to create graphs divided into passages. A passage may include a title and tags, which are useful to categorize it, as well as a content and links to other passages. In Twine, we structured the passages as follows:

- **Social Practices and Frames**: a frame is defined as a passage tagged with context tags (e.g., greeting, morning, formal, hospital) and tags of knowledge acquired during the conversation. As previously stated, connected frames represent the plan patterns of a social practice, which means that frames tagged with the same context tags might appear in different social practices.
- **Cognitive Resources**: a cognitive resource is linked to a frame by its tags (e.g., if a frame is labelled with "greeting, formal", all cognitive resources tagged with these two tags belong to that frame). The title of a cognitive resource passage represents the associated utterance. These passages can be labelled with a role tag based on whether their utterances work for both social actors (no role tag necessary) or just one of them. Cognitive resources can be connected or not to other cognitive resources, being part of small dialogue trees with one or more elements. We can also specify knowledge base updates in these passages.
- **Roles**: the social roles of the user and the agent can be easily switched in a predefined passage tagged with "roles".

Further details on how to edit the dialogue data can be seen in the dialogue system's GitHub page [5]. It is worth noting that we modelled the social practices, frames and cognitive resources ourselves in Twine, tags included, aiming to verify if the dialogue system worked for a specific example (as seen in 4.4).

4.4 Anamnesis Scenario

We created a short anamnesis scenario in Twine, inspired by the social practice in Table 1, defining the roles of the user and the agent (doctor or patient), the timeout variable (30 seconds), the frames ("greeting friendly or formal", "introduction", "gather info", "ask symptoms", "diagnosis", "ask fever or cough", ...), their connections (practices), and their associated dialogue trees (cognitive resources). Then, we ran the dialogue system using this scenario as input.

In Figure 2, the user plays the "patient" role, while the agent plays the "doctor" role. In this case, it is possible to observe the agent changing the frame according to the received user input. For example, when the user reports having a fever (U1), the agent will progress from frame "Fever" to frame "Cough" (A3), following the conclusion of the "Fever" dialogue tree path. As seen in the practices represented in Twine, there are two frames linked after "Fever": "Diagnosis" and "Cough". These have the same frequency, implying that the

[5] https://github.com/GAIPS/socially-aware-interactions

choice of the salient frame should be, presumably, random. However, the agent added to its knowledge base the information that the user had a fever, so, given that the frame “Cough” has a “fever” knowledge tag, its knowledge base match and, consequently, its salience, will be higher.

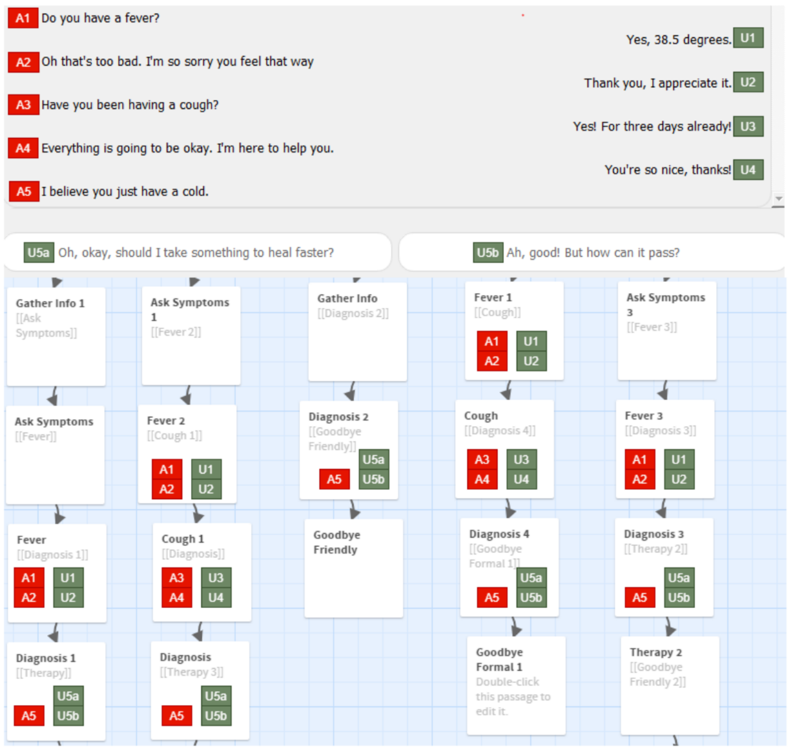

Fig. 2. Chat Interface and Twine - Anamnesis Scenario: Agent (Red Labels) is Doctor and User (Green Labels) is Patient

In Figure 3, the roles of the dialogue entities switch. In this situation, we can see the agent following the user’s lead, continuing on the same frame as the user input frame. Moreover, we can verify that the user options always adapt to the context. For instance, once the dialogue tree associated with the frame “Gather Info” finishes (A4), the options given for the user input are associated with the frames following this frame, such as “Ask Symptoms” (U5a, U5b) or “Diagnosis” (U5c, U5d). Regarding the timeout error (highlighted in blue), there is a moment in which the user is not saying anything, so the agent activates the “Timeout” frame, stating “Is everything okay? You are not replying...”. When the user finally replies, the agent proceeds to repeat the previous utterance, making the conversation return to its previous state, in this case, the frame “Gather Info”.

Fig. 3. Chat Interface and Twine - Anamnesis Scenario: Agent (Red Labels) is Patient and User (Green Labels) is Doctor

5 Discussion

The proposed dialogue system is a hybrid of scripted dialogue tools [11] and natural language dialogue systems [1,4,17], with certain similarities and differences, aiming to address the key concerns of these approaches, while considering their strengths.

In tree-based dialogue tools, authoring dialogue scenarios is very intuitive. A minor issue, however, is that authors must configure a single dialogue tree, which may be overwhelming in large scenarios. On the other hand, natural language dialogue systems lack good authoring mechanisms and thus do not allow scenarios to be easily edited or created for new domains. As a result, authoring was one of the system requirements we addressed, meaning we had to ensure that the dialogue data was structured in an intuitive and easy-to-manage manner, in contrast with the natural language dialogue systems we presented. We illustrated this using Twine, a well-known story-telling tool, in which an author can create and modify dialogue scenarios by altering the context tags and utterances associated with small dialogue trees, as opposed to having one big dialogue tree like in scripted dialogue tools. These trees belong to a collection of tagged frames, which are then organized in practices. Some of these frames and, by extension, their associated dialogue trees, are transversal, in the sense that they can be applied and reused in different scenarios (e.g., greeting, farewell, error handling).

In natural language dialogue systems, users can write their own input, and the agent will respond appropriately, using, for example, a rule-based approach. Contrastingly, in tree-based dialogue tools, users have limited options available. Also, the dialogue is predefined by its author, meaning that the agent does not adapt to changes in the context and instead simply follows the dialogue tree's course. Then, returning to our system, we considered flexibility to be relevant as well, requiring the agent to deliberate on top of the context to exhibit a flexible and appropriate behaviour, like in natural language dialogue systems. Thus, with the agent's knowledge structured as previously described, the agent has to identify the most salient frames, based on the current context, in order to reply appropriately to the user with one of their associated resources. Similar to tree-based dialogue tools, user inputs in our system are not self-made. However, in addition to the agent responses, user options also adapt to the context of the conversation, allowing the roles of the dialogue entities to be easily switched.

Finally, our agent's deliberation mechanism presents robustness as well, another requirement we considered, meaning that the system is able to manage a variety of scenarios and contexts without collapsing, as opposed to natural language dialogue systems, which typically adapt well in one domain, but struggle with new scenarios. On the plus side, natural language dialogue systems usually include mechanisms to detect and handle user errors in the conversation. Considering this, we are also in the early stages of dealing with user errors, but so far we have only handled the "timeout error". Our approach is based on having general frames that deal with unexpected events and that can be activated at any time during the conversation.

In essence, the existing contrast between tree-based dialogue tools and natural language dialogue systems has led us on the search for a compromise between the two. In our system, this middle ground relies on the fact that we have small easy-to-author dialogue trees grouped in tagged frames that can, in turn, be activated according to the context of the conversation, using a general and static deliberation mechanism that functions in a simple, flexible, and robust manner. The presented work, therefore, adds to the development of context-aware conversational agents by proposing a novel approach to organizing the agent's knowledge that meets the requirements of authoring, flexibility, and robustness.

One of our system's main limitations is the fact that it was not tested with users, not only those interacting with the agent, but also the ones configuring the agent's knowledge, being this an essential step for the future. Furthermore, user inputs should be self-made to ensure that they acquire the necessary communication skills for a specific scenario. Consequently, by allowing users to enter their own input, more errors will occur, such as contradictions, repetitions, unrelated information, among others, which will have to be identified and managed by an improved error handling mechanism. Moreover, the structure of the conversation, in terms of its speaking turns, requires more flexibility and should allow dialogues to present not only a simple turn-taking behaviour, but also interruptions and no specified turns to speak.

6 Conclusion

Our work began with the following research question: "Is it feasible to create a flexible and robust dialogue system that adapts to changes in the social context, while maintaining the authoring property of tree-based dialogue tools?". Based on the Cognitive Social Frames model and the theories of Social Practices and Social Identities, we then developed a prototype of a dialogue system that contributes to the creation of context-aware conversational agents. We deemed authoring, robustness, and flexibility to be the most relevant system requirements, having small dialogue trees arranged into interconnected frames, which can be activated by the agent to give an appropriate response according to the current context. With the mentioned requirements met, new scenarios, such as the anamnesis scenario we provided as a demonstration, can be easily configured and played, allowing users to go through a variety of conversation paths. Therefore, this method of structuring the conversational agent's knowledge allows dialogues to be intuitive to configure, while also enabling the agent to adapt to context changes. As future steps, we would like to include users in the system evaluation, translate user inputs to context tags, handle additional dialogue errors, support conversations with unspecified speaking turns, and, finally, create richer scenarios using crowd-sourced data.

Acknowledgement(s)

The presented work was funded by the European Commission as part of the Humane-AI-net project, under grant agreement number 952026. It was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020. Diogo Rato acknowledges his FCT grant (SFRH BD/131024/2017).

References

1. Augello, A., Gentile, M., Dignum, F.: Social agents for learning in virtual environments. In: International Conference on Games and Learning Alliance. pp. 133–143. Springer (2016)
2. Augello, A., Gentile, M., Weideveld, L., Dignum, F.: Dialogues as social practices for serious games. In: Proceedings of the Twenty-second European Conference on Artificial Intelligence. pp. 1732–1733 (2016)

3. de Bayser, M.G., Cavalin, P., Souza, R., Braz, A., Candello, H., Pinhanez, C., Briot, J.P.: A hybrid architecture for multi-party conversational systems. arXiv preprint arXiv:1705.01214 (2017)
4. Campillos-Llanos, L., Thomas, C., Bilinski, É., Neuraz, A., Rosset, S., Zweigenbaum, P.: Lessons learned from the usability evaluation of a simulated patient dialogue system. Journal of medical systems **45**(7), 1–20 (2021)
5. Dey, A.K.: Understanding and using context. Personal and ubiquitous computing **5**(1), 4–7 (2001)
6. Dignum, F., Dignum, V., Prada, R., Jonker, C.M.: A conceptual architecture for social deliberation in multi-agent organizations. Multiagent and Grid Systems **11**(3), 147–166 (2015)
7. Dignum, F., Prada, R., Hofstede, G.J.: From autistic to social agents. In: Proceedings of the 2014 international conference on Autonomous Agents and Multi-Agent Systems. pp. 1161–1164 (2014)
8. Dimas, J., Prada, R.: Dynamic identity model for agents. In: International Workshop on Multi-Agent Systems and Agent-Based Simulation. pp. 37–52. Springer (2013)
9. Hennessy, J., West, M.A.: Intergroup behavior in organizations: A field test of social identity theory. Small group research **30**(3), 361–382 (1999)
10. Holtz, G.: Generating social practices. Journal of Artificial Societies and Social Simulation **17**(1), 17 (2014)
11. Jeuring, J., Grosfeld, F., Heeren, B., Hulsbergen, M., IJntema, R., Jonker, V., Mastenbroek, N., van der Smagt, M., Wijmans, F., Wolters, M., et al.: Communicate!—a serious game for communication skills—. In: Design for teaching and learning in a networked world, pp. 513–517. Springer (2015)
12. Rato, D., Prada, R.: Towards social identity in socio-cognitive agents. Sustainability **13**(20), 11390 (2021)
13. Reckwitz, A.: Toward a theory of social practices: A development in culturalist theorizing. European journal of social theory **5**(2), 243–263 (2002)
14. Tajfel, H., Turner, J.C.: An integrative theory of intergroup conflict. The Social Psychology of Intergroup Relations pp. 33–47 (1979)
15. Trepte, S.: Social identity theory. In: Bryant, J., Vorderer, P. (eds.) Psychology of entertainment. pp. 255–271. Routledge (2006)
16. Turner, J.C., Oakes, P.J., Haslam, S.A., McGarty, C.: Self and collective: Cognition and social context. Personality and social psychology bulletin **20**(5), 454–463 (1994)
17. Weideveld, L.: Social Agents for Learning in Virtual Environments. Master's thesis, University of Utrecht (2017)
18. Zhao, R., Papangelis, A., Cassell, J.: Towards a dyadic computational model of rapport management for human-virtual agent interaction. In: International Conference on Intelligent Virtual Agents. pp. 514–527. Springer (2014)
19. Zhao, R., Sinha, T., Black, A.W., Cassell, J.: Automatic recognition of conversational strategies in the service of a socially-aware dialog system. In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 381–392 (2016)
20. Zimmermann, A., Loarenz, A., Oppermann, R.: An operational definition of context. In: International and Interdisciplinary Conference on Modeling and Using Context. pp. 558–571. Springer (2007)
21. Zimmermann, A., Lorenz, A., Specht, M.: Applications of a context-management system. In: International and Interdisciplinary Conference on Modeling and Using Context. pp. 556–569. Springer (2005)