Ayedoun et al. *Research and Practice in Technology Enhanced Learning* (2023) 18:27
https://doi.org/10.58459/rptel.2023.18027

Research and Practice in
Technology Enhanced Learning

RESEARCH Free and Open Access

An authoring tool for task-oriented dialogue scenarios design in EFL context

Emmanuel Ayedoun [1] *, Yuki Hayashi [2] and Kazuhisa Seta [2]

*Correspondence:
emay@kansai-u.ac.jp
Faculty of Engineering Science,
Kansai University,
3-3-35 Yamatecho Suita,
564-8680 Osaka, Japan
Full list of author information is
available at the end of the article

Abstract

Computer-based conversational environments have been advocated as a promising approach for providing virtual, yet realistic opportunities for communication practice to second language learners. However, the high authoring costs of such environments continue to prevent their widespread diffusion and adoption. Furthermore, there is a limited set of authoring interfaces dedicated to making the creation of dialogue scenarios in the context of language learning easier. In this research, we present a dialogue scenario authoring system that could aid the rapid implementation of desirable dialogue scenarios, lowering the barrier to dialogue scenario authoring for non-programmers or even educators. To that end, we built a pseudo-versatile dialogue scenario authoring interface that enables the automatic generation of services-related dialogue scenarios by leveraging the common underlying structure of services (restaurant, hotel, travel planning, etc.) that appear to share a certain degree of similarity at the task level. Here, we describe the proposed system's features and present the findings of an experimental evaluation study that suggests the usefulness of our approach to facilitating dialogue scenarios designed by people with no prior experience authoring dialogue systems components. According to an evaluation of the tool by a second language teaching expert, the proposed system might also foster second language teaching and learning by allowing both educators and learners to participate in the design of dialogue scenarios that are adapted to different levels of learners.

Keywords: Authoring tools, Dialogue scenario design, Adaptive language learning, Conversational agents, Willingness to communicate in L2

Introduction

The lack of suitable conversation opportunities to practice the target language is a major factor impeding the second language (L2) learners' willingness to communicate (WTC) in the target language. Hence, conversational systems that provide authentic opportunities to simulate daily interactions may be particularly beneficial for second language learners in

terms of improving both their cognitive and emotional communication readiness (Reinders & Wattana, 2014). For instance, to sustainably improve L2 learners' motivation toward communication, it is important to provide them with numerous opportunities to freely simulate and enjoy natural conversation in various realistic dialogue situations. However, this is not a simple task; it requires a careful design of the various dialogue scenarios to the extent of achieving a high degree of reality in interactions that is similar to what learners are likely to encounter in daily face-to-face situations. Furthermore, while dialogue scenarios are an important component of spoken dialogue systems, it has been noted that their design remains largely an art (Farkhadov et al., 2019), owing to the level of degree of creative talent and technical proficiency required for conceptual idea-expression associated with such activity. Therefore, the widespread diffusion of computer-assisted conversational environments has been tempered by the relatively large number of skills and resources required for their implementation.

CEWill (Ayedoun et al., 2019) is an example of a conversational environment that offers L2 learners task-oriented spoken dialogue simulation opportunities in a restaurant context. Dialogue scenarios in CEWill are designed using a knowledge-based approach, allowing it to achieve a deeper level of understanding and control over the conversation flow, increasing the degree of reality of interactions. However, the significant level of knowledge engineering effort and the degree of dialogue expertise required for implementing new scenarios in this system represents a significant barrier that may limit its adoption and frequent use in real educational settings.

To address these issues and promote the availability of a rich pool of realistic dialogue scenarios for L2 learners, this study proposes a dialogue scenario authoring tool that could facilitate the rapid prototyping of desirable dialogue scenarios and contribute to lowering the authoring barrier for non-programmers or educators, who are not necessarily knowledge or software engineers. To that end, we built a dialogue scenario authoring interface that enables the automatic generation of task-oriented dialogue scenarios across various service domains by leveraging the common underlying structure of services (restaurant, hotel, travel-planning, etc.) that appear to share some task-level similarity.

In summary, this study addresses the following three research questions:

- RQ1: Is the authoring interface user-friendly enough for human novices to overcome the difficulties associated with dialogue scenario specification?
- RQ2: Are there any differences between scenarios generated automatically by the system and those designed manually by human novices?
- RQ3: Do the proposed authoring interface and its features help promote second language learning and teaching?

The rest of this study is organized as follows: we begin by comparing our proposed authoring tool to previous works and demonstrating how our contribution is novel and original. Then, we discuss the requirements that should ideally be fulfilled by a suitable authoring tool in the context of this study and indicate how these prerequisites can be addressed. The developed dialogue scenario authoring tool is then introduced and some of its features are described. Following that, we report on a series of experimental evaluations that were conducted to answer the research questions listed above. Finally, we discuss the implications of our findings and present the conclusions as well as future works.

Literature review

Authoring tools and learning support

Several recent reviews have noted the effectiveness of learning support systems and particularly intelligent tutoring systems, emphasizing that well-designed systems can successfully complement or substitute other instructional models in various common academic subjects (du Boulay, 2016; Ma et al., 2014; VanLehn, 2011). However, these tutoring systems continue to be difficult to create. Hence, for the past years, extensive work has been conducted on developing authoring tools to speed up the development of learning support systems, reduce implementation workload, and lower the skill requirements. As a result, several authoring tools such as ASPIRE (Mitrovic et al., 2009), ASTUS (Paquette et al., 2015), AutoTutor tools (Nye et al., 2014), SimStudent (Matsuda et al., 2015) have been proposed and most of them do not require advanced programming. Some authoring tools such as EDUCA (Cabada et al., 2011) also enable the creation of adaptive learning content in the context of intelligent tutoring systems.

Authoring tools for conversation-based learning environments have primarily focused on assisting non-technical users in the creation of pedagogical agent dialogues. AutoTutor (Graesser et al., 2005) provides multi-agent conversational interactions to tutor students using the discourse patterns of a human tutor and has been used in various domains, including computer literacy and physics. To aid subject matter experts in creating dialogue-based tutor authoring tools, such as AutoTutor Script Authoring (Susarla et al., 2003) and AutoLearn (Preuss et al., 2010) have been developed to facilitate the application of AutoTutor to other domains. Similarly, for the tactical language and culture training system (TLCTS), an authoring tool has been created which allows subject matter experts to create pedagogical dialogue for a foreign language learning training system at lower costs (Meron et al., 2007).

However, despite the potential for increased student engagement and the reduced cost of creating lifelike virtual characters, pedagogical agents have not yet achieved widespread adoption in computer-based learning environments (Lester et al., 2015). Although the

authoring tool environments are certainly useful for implementing pedagogical agents for specific domains, they appear to suffer from a lack or limited level of abstraction or versatility of their encapsulated initial domain knowledge, limiting the reusability of their key components across different domains.

Task-oriented dialogue systems and second language communication

The purpose and promise of computer-supported language learning technologies are to facilitate instruction that is personalized to the needs of individual learners (Kerr, 2016). Such systems have been found to be useful in engaging the learner in the educational experience (Conlan et al., 2007). To sustainably enhance L2 learners' willingness to communicate, previous research has emphasized the importance to provide learners with various realistic opportunities to simulate conversation using the target language.

Interestingly, it has been suggested that task-oriented dialogue systems, where a task should be accomplished in the target language, have a clear potential for placing the student in a realistic dialogue situation (Raux & Eskenazi, 2004). Building on such views, in our previous works, Ayedoun et al. proposed CEWill, an embodied conversational agent that provides L2 learners with opportunities to freely simulate spoken dialogue in realistic daily-life settings such as talking with a waiter in a restaurant (Ayedoun et al., 2019). The system, which interface is shown in Figure 1, required a carefully handcrafted dialogue scenario for each situation and was equipped with a set of domain-independent conversational strategies aiming to foster the system's ability to carry on smooth and warm interactions with learners. This was achieved by equipping the conversational agent CEWill with two types of conversational strategies (i.e., Communication Strategies, and Affective Backchannels), allowing it to cover both aspects related to communicative breakdowns that often occur in L2 learners-agent interactions and those related to affective variables influencing willingness to communicate, in accordance with MacIntyre's model (MacIntyre et al., 1998). By enabling the dialogue agent to make use of Communication Strategies, the goal is to enhance the agent's strategic competence, which frees learners

Fig. 1 CEWill interface and learner interacting with the agent Peter in restaurant scenario (Ayedoun et al., 2019)

from the challenging burden of resolving communication pitfalls by themselves. Through the use of Affective Backchannels, researchers aimed to foster the CEWill's ability to convey empathetic and WTC-friendly support to L2 learners. Results of experimental evaluations demonstrated the meaningfulness of such a conversation simulation environment, especially in countries where English learning focuses less on the development of communicative skills and where learners have limited access to opportunities for using the target language in authentic settings. However, in addition to conversational scaffolds such as Communication Strategies and Affective Backchannels, a careful design of the dialogue scenario itself is paramount to achieve a high degree of reality in interactions similar to what learners are likely to experience in daily face-to-face situations. Yet, it seems important to bear in mind that the design and implementation of a dialogue scenario are not an easy undertaking that requires a certain degree of expertise and knowledge about scenario design and dialogue systems. Hence, this important requirement may serve as a barrier to the widespread large diffusion and adoption of conversation simulation environments such as CEWill.

Novelty and contribution

In the light of the contributions and limitations of the different studies described above, we note that although several authoring tools for dialogue systems have been proposed in the artificial intelligence in education community, the majority of these systems were designed to carry on tutorial dialogues in specific subjects such as physics, computer literacy, or critical thinking. Tutorial dialogues achieved within such systems deal with replicating dialogue moves of human tutors in teaching situations, which are dialogue scenarios quite different from task-oriented dialogue scenarios targeted in the context of the current work.

In contrast, because of the knowledge-based approach used in their implementation, conversation interfaces such as CEWill and Enskill (Johnson, 2019), that provide L2 learners with realistic conversation opportunities, can achieve a deeper level of understanding and control of the conversation flow, increasing the degree of reality of interactions, as described in the previous section. However, the disadvantage of such systems is that for new scenario implementation, scenario designers must handcraft the dialogue task structure and the dialogue flow logic from scratch. Additionally, since the dialogue scenario contents are viewed here as the learning material, scenario designers may be required to consider learners' proficiency or needs for effective dialogue scenario design. Such constraints even further complicate the scenario design task, which can quickly become a time- and resource-consuming activity requiring not only significant knowledge-engineering effort but also some level of expertise in dialogue systems on top of proper pedagogical knowledge.

When confronted with the aforementioned issues, we discovered the lack of a generic tool that is user-friendly enough to allow non-programmers to quickly create suitable task-oriented dialogue scenarios.

Hence, the major contributions of this study are twofold:

- an authoring framework for rapid implementation of a relatively broad range of situational dialogue scenarios to lower the dialogue scenario authoring barrier for non-programmers or educators who are not necessarily knowledge or software engineers. This is accomplished through the use of a task structure-based authoring approach, which enables us to target a suitable subset of dialogue domains that share a coherent structure at the task level and exploit such a shared task structure to implement a versatile authoring tool that embeds domain-independent reusable dialogue task components. Therefore, the proposed system makes it easier to design various dialogue scenarios that fall under the hood of the targeted common task.
- an approach to automatically generate dialogue scenarios (i.e., learning contents) that might be tailored to the needs or proficiency of L2 learners. This is accomplished by incorporating a feature that allows scenario designers to generate both basic and advanced versions of the same dialogue scenario without requiring extensive additional content authoring effort.

To the best of our knowledge, no dialogue scenario authoring framework currently aims for or can achieve both of these aspects in the context of language learning in general, and particularly in the context of L2 learning.

Research objective and requirements

Research objective

Murray (1999) alluded to the difficult trade-off issue of how much the difficult task of authoring learning support systems could be scaffolded: ideally, a desirable authoring tool should be both specific enough to make authoring template-based, and general enough to be appealing to many educators. According to Woolf (2010), the complexity of authoring tool development is also due to design tradeoffs (specificity vs. generality). Based on these assumptions, the current study proposes a flexible authoring environment that could facilitate the design of dialogue scenarios for a relatively broad range of different conversation situations while still incorporating a certain level of precise knowledge about the target domains. If achieved, such a balance between authoring ease and tool flexibility will allow scenario designers (i.e., educators) to have more control over the specification of desirable dialogue contents, which is a crucial requirement for conversational applications in the field of education. This would ultimately promote the availability of a

diverse pool of dialogue scenarios for L2 learners, which would be beneficial in terms of increasing their motivation toward communication.

One of the driving concepts behind the creation of this authoring interface is the use of a task-oriented authoring approach, which allows us to target a suitable subset of dialogue domains that share a coherent structure at the task level and exploit such common task structure to implement an authoring tool that embeds pseudo-domain-independent reusable dialogue task components. We anticipate that doing so will make the design of various dialogue scenarios that fall under the hood of the targeted common task easier. In this study, our goal is specifically to implement a system that can serve as a proof of concept of the feasibility and usefulness of our approach, despite the complexity of such a challenge. In the present work, our goal is specifically to implement a system that can serve as a proof of concept of the feasibility and meaningfulness of our approach, despite the complexity of such a challenge.

Requirements and their fulfillment

Achieving a high level of versatility while keeping authoring costs low for content creators is a difficult task, but it is certainly necessary to enable more active and frequent use of authoring tools and learning support systems by educators who are not necessarily software engineers. Keeping this in mind and drawing inspiration from Murray's review of authoring tools (Murray, 1999), we identified several core requirements for the desired dialogue scenario authoring tool in the context of this study. These prerequisites are described in the following lines, along with how they are addressed in the built tool.

Requirement 1 (embed a relevant level of domain-independent knowledge about the dialogue task structure): this refers to some generic knowledge about the common structure of various dialogue domains that the system will target. Such knowledge, if pre-wired and embedded in the tool, could make authoring easier and more powerful by allowing the same structure to be reused across multiple instances of dialogue domains. Therefore, dialogue scenario authors can concentrate on specifying domain-specific aspects of the dialogue flow, significantly reducing new scenario implementation efforts and contributing to decreased authoring time and costs. This could assist developers with no or limited programming skills in designing and cost-effectively implementing new conversation scenarios, primarily through the specification of properties for key task components in the target domain.

To meet Requirement 1, for covering various dialogue domains such as *Restaurant*, *Hotel* and *Transport*, is to embed a domain-independent task model shared by all these different domains in the authoring tool. To achieve this, we chose the services model (Ferrario et al., 2011) because domains related to services appear to share a certain degree of structural

similarity at the task level. In other words, building an authoring tool that embeds a generic model of services could make it easier to implement services-related dialogue scenarios in various contexts, such as hotels, restaurants, or transport. This approach is expected to allow a scenario author to create various dialogue scenarios without having to worry about defining the task structure from scratch. Furthermore, focusing on dialogue domains related to services may provide L2 learners with a wealth of desirable communicative practice since they are likely to face such dialogue situations in their daily lives.

Requirement 2 (make possible efficient authoring flow and knowledge management): this involves the authoring environment's ability to scaffold the dialogue scenario specification by allowing input through the use of templates, data entry forms, pop-up menus, and so on.

To meet this requirement, scenario authors in our proposed authoring tool are allowed to select rather than type whenever the range of possible input values can be limited to a finite set. Furthermore, we made a clear distinction between the various types of information that scenario authors must deal with (i.e., actions specifications, interface parameters) and those that the system handles discretely behind the scenes (goals and actions structure generation, dialogue flow generation, interface parameters) as we will explain in the following section. This is expected to help lower the skill threshold for dialogue scenario modeling and allow actual educators and others with no or little programming experience to participate in the dialogue scenario specification process.

Requirement 3 (enable scenarios designers to apprehend the structure of authored dialogue scenarios): authoring tool usability frequently comes at the expense of expressiveness (Engström et al., 2018). This necessitates making the interface expressive and accessible enough to scenario designers so that they can reflect on their scenario design, assess whether it meets their expectations, and make changes if necessary.

To meet Requirement 3, our proposed authoring tool includes a user-friendly graphical interface that allows scenario designers to examine both the static structure of the designed dialogue scenario (later shown in Figure 5(3)) and the dialogue control dynamics across possible dialogue paths (later shown in Figure 6). This allows scenario designers to participate in the authoring process interactively, examine the authored dialogue structure, validate generated scenarios, and make any necessary changes to achieve the desired dialogue scenario. This function may be particularly useful for reducing the cognitive load associated with the creation of complex dialogue scenarios.

Requirement 4 (facilitate content modularity, customization, and reusability): this refers to the authoring tool's ability to allow modular authoring of the different components

needed to design a desired dialogue scenario and their storage as library structures so that they can be reused for multiple scenarios purposes or for adjusting to different learners' level. This may also facilitate the diffusion of the principles of shared dialogue scenario design for studies dealing with communicative aspects of L2 acquisition. For instance, input processing, output processing, conversational strategies, etc. should be encapsulated in sub-components with well-defined interfaces that are decoupled from the dialogue scenario itself.

As far as the fulfillment of Requirement 4 is concerned, the dialogue scenario generated by the current authoring tool is intended to be fully compatible with conversational agent systems such as CEWill, eliminating the need to implement from scratch other key components dedicated to dialogue management, natural language processing, and agent animation generation. The discussion related to the implementation of such important modules is thus omitted voluntarily from this paper to focus emphasis on how we can support the dialogue scenario authoring process.

Conceptual framework for dialogue scenarios generation

From generic model of services to dialogue scenarios

To facilitate the authoring process, the system incorporates a generic model of services based on the conceptual framework proposed in a previous publication (Ayedoun et al., 2020). This framework was inspired by Ferrario's prior work (Ferrario et al., 2011) on service conceptualization. According to Ferrario and his colleagues, despite the term service's pervasiveness in everyday discourse, it is used in different ways across disciplines, and even within the same discipline, confusion and inconsistencies are common. They further mentioned that in such a situation, service designers may lack a common semantic background, which could harm service interoperability. To overcome this issue, they proposed a high level of abstraction to propose a generic (ontological) model of service structure to facilitate a unified understanding of concrete services and their facets. For instance, according to their definition, service is present at a time *t* and location *l* if, at time *t*, an agent is explicitly committed to guaranteeing the execution of some type of action at location *l*, on the occurrence of a certain triggering event, in the interest of another agent and upon prior agreement, certainly (Ferrario & Guarino, 2008). Moreover, concerning the service delivery, they proposed that the central part of the service process is given by the customized service content delivery, which is the actual event in which one executes what has been promised in the service agreement; it is composed of the following:

- **Core service Actions:** are those actions whose execution contributes to meeting customers' needs. In a sense, these actions characterize service for what it is and must

necessarily be exposed to the customer, such as the action of *serving food or drink* in a restaurant.

- **Supporting Actions:** are actions required for the service but are not explicitly mentioned as constituting the service, such as *guiding the customer to a seat or explaining the menu* in a restaurant.
- **Enhancing Actions:** these are the actions designed to augment the value of the service. These actions can be considered as additional service actions that are related to but not strictly included in the main service, such as *providing karaoke*, *massage*, or any other form of entertainment as an option in a restaurant.

We can conclude from the foregoing that any service-related dialogue scenario can be expressed as a combination of these three key components (Core Actions, Supporting Actions, and Enhancing Actions). We use this conceptualization of the service system process as proposed by Ferrario and colleagues as a baseline for our work, and further drill into the structure, properties, and relationships of these three types of service actions to reuse the same generic components in different service-related dialogue scenarios. For a more in-depth description of the service system process, refer to Ferrario's article (Ferrario et al., 2011).

We consider that a service-related dialogue task structure should ideally capture the goals and actions that will be performed to achieve the delivery of the target service. Thus, the Service Goal and the Service Actions are two key components of the dialogue Task Structure. To better illustrate how the service goal and service actions are structured, we present an example of a service-related dialogue scenario that most people should be familiar with following: the restaurant scenario, where a customer walks into a restaurant, gets guided to a table and orders drinks and foods before settling the bill and leaves.

The Service Goal is composed of initial and final states of the service. It is expressed in terms of customer Needs, and spatio-temporal requirements within which the Service delivery occurs. In the example shown in Figure 2, the service is intended to satisfy the customer Needs (*Drink* and *Food*), with the interaction starting at the *Entrance* of the restaurant, and ending at the *Cashier* position.

Fig. 2 Example of Service goal

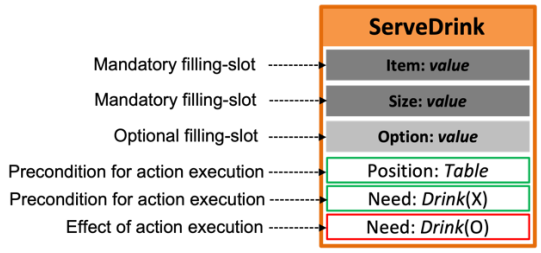

Fig. 3 Example of Service action structure

The Service Actions are designed to move the interaction from its initial to final states as defined by the service goal. Their execution triggers the actual dialogue between the service customer (e.g.: the learner) and provider (e.g.: the system). We propose that the internal structure of each Service Action is composed of mandatory and optional slots that must be filled through actual dialogue with the customer to enable the execution of a given Service Action. In addition, a Service Action can only be executed when certain prerequisites on the interaction state are satisfied. Similarly, the execution of a given Service Action may update the interaction state through fulfillment of the customer's Need, or the spatio-temporal restriction. In the example provided in Figure 3, the Service Action *ServeDrink* action can only be performed when the current Position is *Table* and the customer Need *Drink* has not yet been satisfied. Furthermore, it requires the necessary information to fill the slots *Item* and *Size* to be executed, while the slot *Option* is defined as optional. Finally, the completion of this Service Action has the effect of fulfilling the customer's Need for a *Drink*.

In summary, using our proposed approach, the scenario designer specifies a hierarchically-organized service task similar to a type of plan structure defined in AI planning, for covering a topic during the authoring process. Each dialogue scenario addresses high-level goals (i.e., Needs to be satisfied) and is generated by the system as a sequence of any combination of Core service Actions, Supporting Actions, and Enhancing Actions based on the designer's specifications.

Authoring workflow for dialogue scenario generation

The scenario authoring workflow provides an overview of the various activities that are conducted to specify the properties of the dialogue task structure, which will result in the generation of the dialogue scenario. As shown in Figure 4, the dialogue task structure specification of a dialogue scenario in a given service domain can be performed by conducting only four activities on the scenario designer side. These activities correspond to those shown in parallelogram shapes i.e., (1), (3), (5), (7) in Figure 4, while other

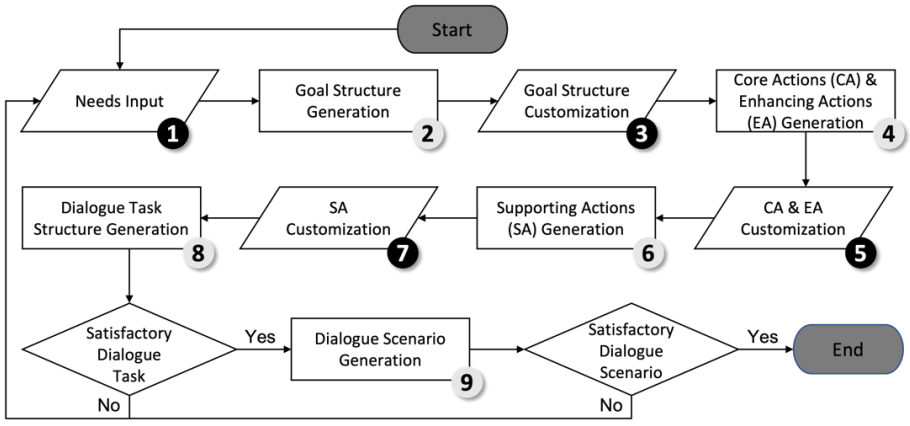

Fig. 4 Dialogue scenario authoring workflow showing the different activities that are carried out for dialogue scenario generation both on human side (parallelograms) and machine side (rectangles)

activities (shown in rectangle shapes, i.e., (2), (4), (6), (8) (9)) are executed behind the scenes by the authoring tool itself.

Note that during the process, the scenario designer is only required to specify execution constraints for each service action through customization of default template structures generated by the system. This simplifies the authoring process for scenario designers while also relieving them of the need to consider the execution flow (i.e., how several actions should combine to form a viable dialogue path), which is rather handled behind the scenes by the system using generic relations derived from the embedded Service model. This hybrid approach is expected to consequently reduce the authoring effort by allowing scenario authors to focus solely on domain-dependent aspects of the target service-related dialogue scenarios, while the authoring tool takes advantage of the common underlying structure (i.e., Service model) to manage inter-domain commonalities. This authoring policy is particularly important in the context of this study since our target users (i.e., scenario designers) are not necessarily familiar with programming or dialogue task design.

After the scenario designer has completed the four activities mentioned earlier, the authored dialogue task structure for the target dialogue scenario is displayed as a hierarchical structure, allowing the scenario designer to visualize and obtain a big picture of the authored domain knowledge. This is the semantic representation of the target scenario or the dialogue task structure. The scenario designer can revisit the relationships between customer Needs and Service Actions to ensure that intended service delivery is achieved. For instance, the scenario designer may decide to revisit the authoring activities described in Figure 4 and make appropriate revisions, add missing components, or even define new customer needs. After the dialogue task structure has been validated by the

scenario designer, the system automatically generates the dialogue scenario, which can be viewed as a finite state machine representing several possible dialogue paths or combinations of Core Actions, Supporting Actions, and Enhancing Actions.

Advantages of the authoring tool from a language learning support perspective

Designing the dialogue scenario as a combination of three types of service actions may provide much flexibility in terms of generating dialogue content that is personalized to the needs of learners from the perspective of L2 learning support. For instance, dialogue scenarios including only Core service Actions and their Supporting Actions (i.e., basic scenarios) may be used for beginners, while complex scenarios containing a combination of basic scenarios and Enhancing Actions may be presented to advanced learners. This way, scenario designers can reuse the same dialogue task structure to generate dialogue content that is tailored to learners' levels without requiring extensive additional content authoring effort.

Additionally, we created a convenient feature that allows scenario designers to "effortlessly" reshape the original dialogue scenario to automatically generate a variant dialogue scenario. We achieved this by using one of the underlying generic properties of the Service task, namely, the Payment Style. For instance, we implemented two Payment setting styles (*PostPay* and *PrePay*) in the system. All scenarios are generated with the *PostPay* mode by default, but upon selection, the authoring tool can also generate a dialogue scenario corresponding to the *PrePay* style. This feature could be useful for enabling learners to practice again a previous dialogue situation from a slightly different perspective (i.e., *PostPay* vs *PrePay*). We will provide concrete examples regarding these features in the next section.

Authoring service-related dialogue scenarios

In this section, we use screenshots from the authoring interface to explain how the tool actively supports the authoring process by guiding and facilitating it behind the scenes. We also provide an additional explanation of the relationships among concepts of Needs, Service goals, and Service Actions as well as Dialogue Task Structure and Dialogue Scenario. Figure 5 shows various windows of the built authoring tool. The system was created as a web application that is accessible through a browser. It has several windows including: the dialogue task specification window (Figure 5(1)), the slot customization window (Figure 5(2)), the dialogue task structure visualization window (Figure 5(3)), and the dialogue scenario visualization window shown in Figure 6.

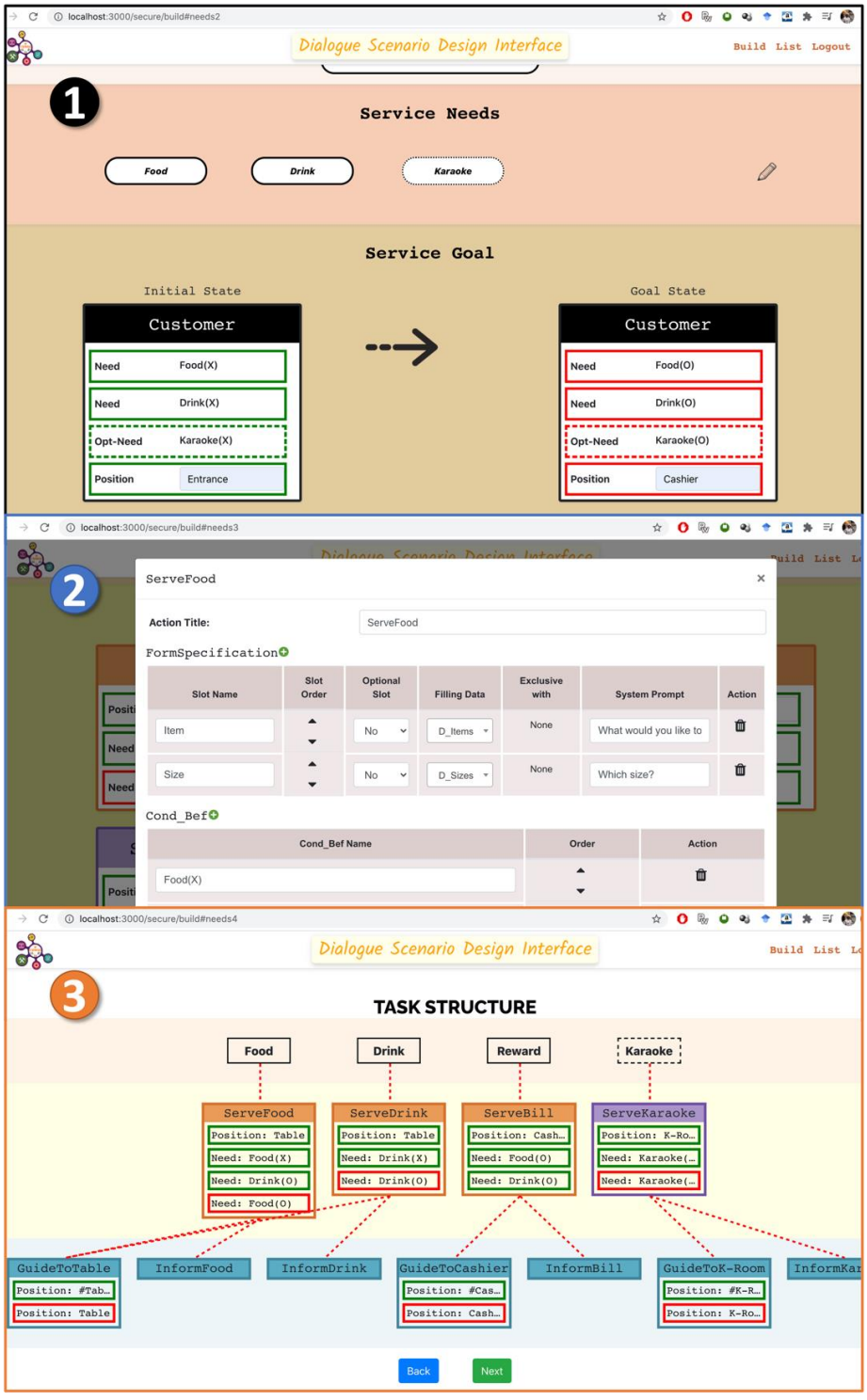

Fig. 5 Authoring tool interface showing different windows corresponding respectively to the dialogue task specification at the macro-level (1), micro-level (2) and the dialogue task structure (3)

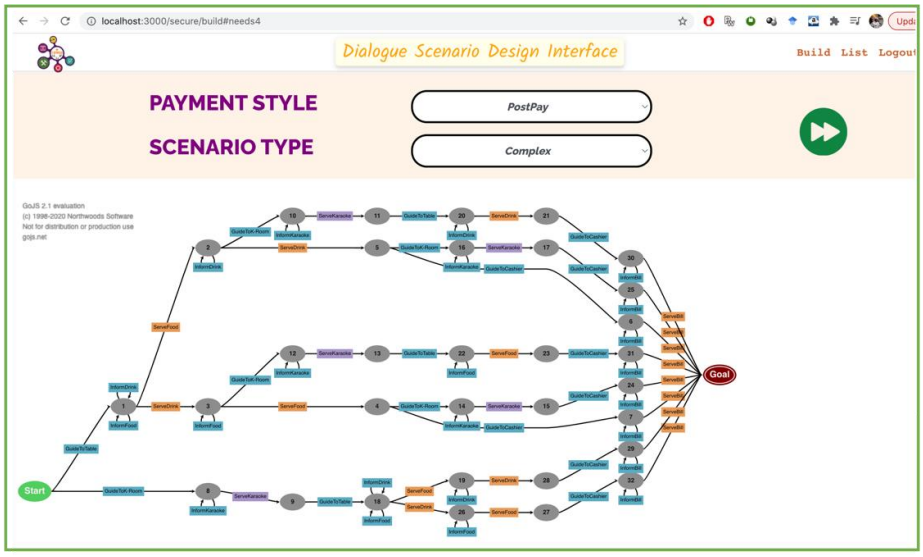

Fig. 6 Authoring tool interface showing an example of automatically generated dialogue scenario in which the nodes representing different dialogue states. These states are connected through edges which are services actions. The blue labels represent Supporting Actions, orange labels represent Core Actions while purple labels stand for Enhancing Actions. A sequence of nodes and edges going from the Start node and ending at the Goal node is called a dialogue path

Dialogue task specification at macro-level

The first step in the authoring process consists of the scenario author specifying the key components (i.e., service actions) of the targeted dialogue task structure.

To begin, the author inputs the customer's Need(s) to be satisfied by the target service, as shown in Figure 5(1). This corresponds to the activity (1) in Figure 4. For instance, in a restaurant scenario, *Drink* and *Food* may be set as primary Needs and *Karaoke* as an optional Need. According to these Need(s), the tool generates the service goal, which is to represent the initial and goal states of the targeted service task. Based on the automatically generated service goal, the scenario author may choose to refine both the initial and goal states by specifying some spatio-temporal requirements or modifying the desired starting and ending state criteria for the interaction. For example, the scenario designer may include a spatio-temporal requirement (i.e., Position) for the service delivery, which can have different values at the start (e.g.: *Entrance*) and end (e.g.: *Cashier*) states. This service Goal refinement allows scenario authors to see the big picture of the target service delivery process and corresponds to the activity (3) in Figure 4.

Following this, the tool generates base structures of Core and Enhancing Actions to be executed to satisfy Need(s) specified in the service Goal. The appropriate types of actions

are automatically selected based on the nature of the target Need(s). The Core Action template structure is generated for each primary Need, whereas the Enhancing Action template structure is set up for each optional Need. The scenario designer can set additional spatio-temporal requirements for each action based on the automatically generated basic structure of Core and Enhancing Actions. Furthermore, the scenario designer is requested to define the slots that must be filled to make possible delivery or execution of the target Core or Enhancing Actions during the interaction. To help the handling of this important task, the scenario designer is prompted with a slot specification window, as shown in Figure 5(2). This corresponds to the activity (5) in Figure 4.

Moreover, basic structures of Supporting Actions are automatically generated and attached to each defined Core Action and Enhancing Action, as shown in Figure 5(3). This is done because Supporting Actions are used to facilitate the delivery of Core and Enhancing Actions. For instance, two types of Supporting Actions are generated: State-transition Supporting Action and Information Supporting Action. State-transition Supporting Actions based on state-temporal restriction are only generated only when the parent Core or Enhancing Action has some spatio-temporal requirement defined. For example, in Figure 5(3), *GuideToTable* is generated for the Core Actions *ServeDrink* and *ServeFood* since the state-temporal restriction Position*: Table* was specified for these actions. Note that, by default, these Supporting Actions inherit the constraints (i.e., spatio-temporal requirements and prerequisite constraints) associated with their parent Core or Enhancing Action. This is because, for a given Supporting Action to be executed, the dialogue state must first be aligned in such a way that the parent Core or Enhancing Action is executable. Note that this feature is intended to relieve scenario designers from the need to specify execution constraints for Supporting Actions on their own. However, these default settings can still be tailored to the characteristics of the target domain. This corresponds to the activity (7) in Figure 4.

From the foregoing, one might conclude that service actions are intended to move the interaction from its initial to final states, as specified in the service goal. The execution of each service action can also be regarded as a catalyst for actual dialogue between the service customer and the provider (i.e., the system and the learner). Furthermore, although aspects of natural dialogue generation and recognition are beyond the scope of this study, the internal structure of each service action is composed of mandatory and optional slots that must be filled through actual dialogue with the customer (i.e., the learner) to enable the execution of a given service action. Additionally, a service action can only be executed when some preconditions or requirements on the interaction state are satisfied; similarly, the execution of a given service action may update the interaction state, as shown in Figure 3.

Dialogue task specification at micro-level

As shown in Figure 3, each Service Action has a certain number of slots. In terms of actual dialogue flow, the execution of each service action can be viewed as a slot-filling driven dialogue management where dialogue slots are progressively filled through actual conversational moves between the service provider and the customer (i.e., the dialogue agent and the learner). Although some slots are already predefined for convenience, the scenario designer can still add new ones or customize existing ones by specifying several properties that will determine the flow of slot-filling during actual dialogue. To help with this important task, the scenario designer is prompted with a slot specification window, as shown in Figure 5(2). This can be regarded as a part of activities (5) and (7) in Figure 4.

Slots properties that are customizable include:

- **Order property**: indicates the order in which slots must be filled to make the dialogue sound more natural. For instance, in the example presented in Figure 5(2), it may be reasonable to fill the slot Item before the slot Size since different sizes (e.g., Small, Medium, Large) options may be available depending on the type of item (e.g., Beef steak, Caesar Salad, Pizza) that is selected.
- **Optional property**: indicates whether the slot's value is indispensable or not for the execution of the target service action.
- **Filling data property**: constrains the semantic type of the target slot. The configuration is conducted by selecting the appropriate type from the predefined options. Scenario authors can still modify existing slot types according to the restrictions of the target domain, or create new types from scratch conversely, the scenario designer can indirectly specify here which types of utterances (i.e., linguistic knowledge) can be accepted to fill in a given slot.
- **Exclusive with property**: shows mutual exclusivity relationships between two slots. This property can be useful in cases where the filling of a given case indirectly allows the filling of another so that to avoid slot-filling redundancy.
- **System prompt property**: shows the system prompt for triggering the learner's response and filling the target slot. For example, when the target slot is *Item* under the Core service Action *ServeDrink*, the system prompt could be specified as "*What would you like to drink?*" or "*Anything to drink?*". Hence, through this slot, the scenario designer can also specify the system's linguistic prompts to constrain the level of linguistic knowledge used by the system when replying or asking something from learners.

Note that both the system prompt property and the filling data properties allow the scenario designer to embed the desired level of linguistic knowledge for a given scenario.

Generation of dialogue task structure and dialogue scenario

The authored dialogue task structure for the target dialogue scenario is displayed as a hierarchical structure, allowing the scenario designer to see the big picture of the authored domain knowledge. As shown in Figure 5(3), the dialogue task structure can be seen as a static task representation of the dialogue scenario. The scenario designer can revisit the relationships between customer needs and service actions to ensure that the intended service specification is achieved. For instance, the scenario designer may decide to make appropriate revisions by adding missing constraints, or even defining new customer needs, if necessary.

Once the dialogue task structure has been validated by the scenario designer, the resulting dialogue scenario or task execution flow is automatically generated by the system, as shown in Figure 6. This can be seen as the collection of all possible task execution paths or dialogue paths concerning constraints specified by the scenario designer through the task specification activities shown in Figure 4. The tool generates the dialogue scenario as a finite state machine of Service Actions, with each state (i.e., node) corresponding to any milestone between the initial and the goal states; arcs (i.e., edges) connecting different states represent service actions; blue labels represent Supporting Actions, orange labels represent Core Actions while purple labels represent Enhancing Actions. A sequence of nodes and edges that starts at the Start node and ends at the Goal node is called a dialogue path. For instance, in the example shown in Figure 6, the system generated several dialogue paths, including dialogue paths involving only Core Actions (i.e., *ServeFood*, *ServeDrink*) and corresponding Supporting Actions (e.g., *InformDrink*, *GuideTable*), and also longer dialogue paths that included an Enhancing Action (i.e., *ServeKaraoke*). Using the information displayed in this window, the scenario designer can visually determine the order in which each service action should be executed as the dialogue unfolds between the agent and the learner. Undesirable or redundant dialogue paths can be removed, and missing ones can be added by revisiting the specifications in the task specification windows (Figures 5(1) and 5(2)). In other words, the task execution flow visualization window serves as a debug interface for scenario designers, allowing them to review dialogue paths generated by the system and make changes as needed. This is an important feature because, although perfectly aligned with the task execution constraints, some of the dialogue paths generated by the system, may sound uninteresting from a communication practice standpoint. Therefore, scenario designers must be able to remove such paths when needed.

It is also worth noting that, at this point, the scenario designer can generate additional variants of the displayed dialogue scenario by switching the Payment Style to *PrePay* or the Scenario Type to *Basic*, as previously explained in the section dealing with Additional language learning support features. In total, four different combinations of Payment Style & Scenario Type (i.e., *PostPay* & *Basic*, *PostPay* & *Complex*, *Prepay* & *Basic*, *PrePay* &

Complex) can be set to generate four different scenarios without having to repeat the dialogue task specification activities. Figure 7 shows examples of various setting combinations and the resulting dialogue scenarios.

Furthermore, when a given dialogue path is selected, the system displays a pseudo-dialogue script corresponding to that path, as shown on the right-hand side of Figure 8. This visualization feature enables the scenario designer to gain a better understanding of the idea of the conversation flow that will occur between the learner and the system along the selected path. This might also be useful for scenario designers who want to debug the generated dialogue scenario thoroughly.

When the scenario designer is satisfied with the generated dialogue scenario, the entire specification data can be saved in XML format for further editing or exported for integration into a conversational agent environment such as CEWill.

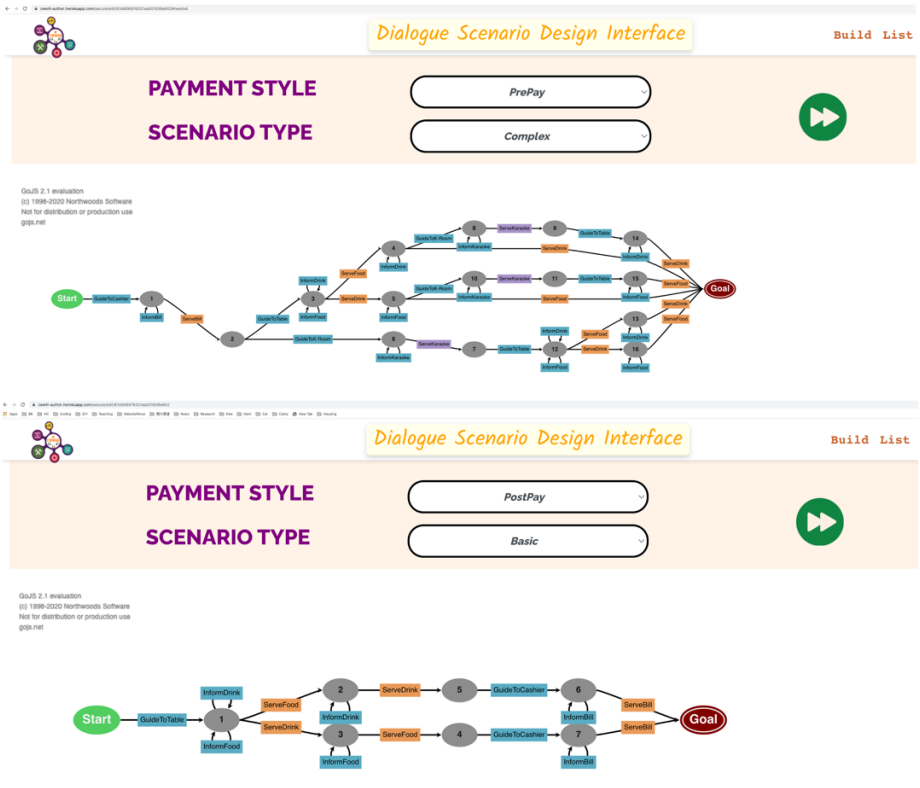

Fig. 7 Authoring tool interfaces showing two automatically generated scenarios for the same domain as in Figure 6 (top: *Prepay & Complex*, bottom: *PostPay & Basic*)

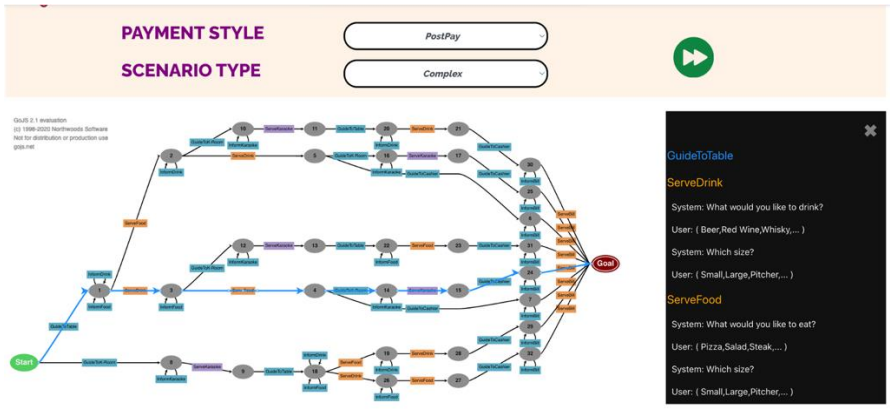

Fig. 8 An example of pseudo-dialogue flow (right) generated for the dialogue path highlighted in blue

Reusability of dialogue scenarios components

To enable the reusability of generated dialogue scenarios in various dialogue domains, we implemented a feature that allows scenario designers to access and edit dialogue scenarios created by others via a dedicated window shown in Figure 9. For instance, a scenario designer through this window can copy or download an existing dialogue scenario and reuse defined service actions, fine-tune or modify them for serving another service-related domain. For example, specific service actions (e.g.: *ServeFood*, *ServeDrink)* from a restaurant dialogue scenario can be edited and reused in a hotel dialogue scenario to create an in-room dining-related dialogue. That is, existing dialogue scenarios can currently be reused within the proposed authoring interface to target different dialogue domains.

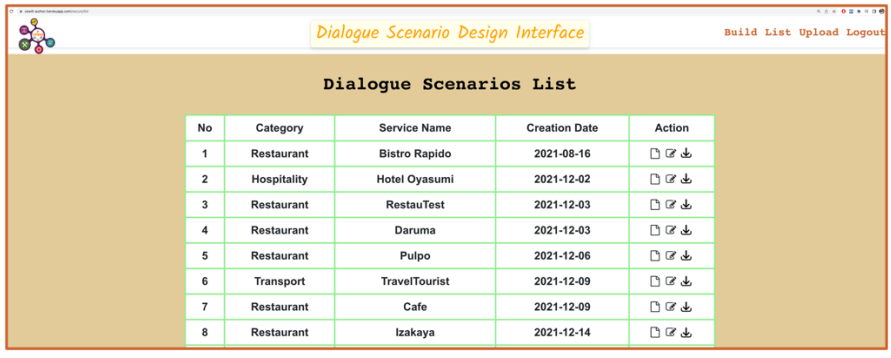

Fig. 9 Authoring tool interface showing an example of dialogue scenarios list through which existing dialogue scenarios can be edited or downloaded to be reused in different domains

Experimental evaluations and results

Objective of the experiments

We conducted a series of experiments to determine how well the authoring tool helped in the design of service-related dialogue scenarios. Our goal was to assess the authoring tool's usability and investigate its effectiveness in terms of reducing authoring difficulty and time while maintaining the quality of generated dialogue scenarios. Additionally, we also evaluated the authoring tool's potential for language teaching and learning support. In short, the design of our experiments was motivated by the following research questions (RQ1~RQ3), as previously stated in the Introduction section:

- **RQ1:** Is the authoring interface user-friendly enough for human novices to overcome the difficulties associated with dialogue scenario specification?
- **RQ2:** Are there any differences between scenarios generated automatically by the system and those designed manually by human novices?
- **RQ3:** Do the proposed authoring interface and its features help promote second language learning and teaching?

Methodology

To answer the aforementioned research questions, we evaluated the dialogue scenario design process both with and without the tool and investigated for any differences in terms of authoring time and difficulty (RQ1). We also evaluated and compared the quality and preference ratio of dialogue scenarios created by human novices (handcrafted dialogue scenarios) to those generated by the system (RQ2). Finally, we consulted an English as a Foreign Language (EFL) teaching expert to evaluate the system's potential to foster L2 teaching and learning (RQ3). Note that the term "human novices" refers to people who are not familiar with dialogue scenario design or have no technical experience designing dialogue systems.

The experimental evaluation was divided into two phases to answer each of our three research questions: Phase 1 (RQ1–RQ2) and Phase 2 (RQ3).

Phase 1

In phase 1, data were collected from a total of twenty-four (24) undergraduate and graduate students over two rounds of experiments with seven (7) and seventeen (17) participants, respectively. Figure 10 shows participants using the authoring tool during the experiments. Results of the first round of experiments were reported in a conference paper (Ayedoun et al., 2021) and the second round of experiments was conducted to increase the sample size and thus improve the reliability of our results. We also rigorously ensured that the same

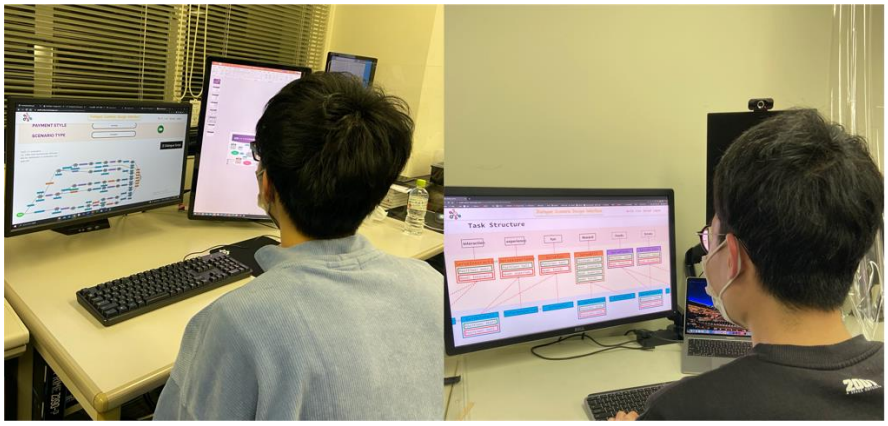

Fig. 10 Two participants during the experimental study

experimental settings were used in both rounds of experiments to maintain uniformity of conditions. All the participants were majoring in engineering and had no previous experience of authoring dialogue scenarios. They were told that their participation in the study was entirely voluntary and that the results would be anonymized. Each of the two rounds of experiments was conducted following the five steps described below:

Step 1 (initial guidance): at the beginning of each session, we provided all the participants with a short set of training materials and training tasks, which we guided them through, to the extent to let them understand what a dialogue scenario is, the key concepts associated with the design of Service-related dialogue scenarios, as well as the authoring tool itself. This required about 30-45 minutes, including Q&A time.

Step 2 (dialogue task specification): participants were asked to use the authoring tool to create a dialogue scenario on their own. We instructed them to target a dialogue scenario involving a service provider and customer in a restaurant context to ensure that the level of task difficulty was homogeneous among participants since it is a service-related dialogue situation that even human novices should be reasonably familiar with. After allowing participants to freely specify the key components of their dialogue task structure (activities 1–8 in Figure 4) using the authoring tool (Figure 5(1–3)) we asked them to stop the authoring process one step before they could visualize the automatically generated dialogue scenario.

Step 3 (handcrafting of dialogue scenarios): we asked participants to handcraft (i.e., manually design) the dialogue scenario that satisfies the constraints of the Step 2 dialogue task. To that end, all participants were provided with a PowerPoint template sheet with samples of finite state machine symbols representing the three types of Service Actions, edges, and nodes used by the authoring tool to generate dialogue scenarios. We gave them as much time as they needed and allowed them to double-check the task structure they had previously specified at the interface. They were also instructed to consider all possible

combinations (i.e., dialogue paths) to make the dialogue scenario the most comprehensive as possible. Figure 11 (top) shows an example of a participant's handcrafted dialogue scenario during the experiment.

Step 4 (comparison of both dialogue scenarios): we allowed them to visualize the dialogue scenario generated by the system and thoroughly compare it with their handcrafted scenarios to find similarities and differences once they were satisfied with their handcrafted dialogue scenario.

Step 5 (evaluation survey): at the end of the experiment, we administrated an evaluation survey questioning their perceived difficulty of the manual generation (handcrafting) of the dialogue scenario, to answer RQ1, and their opinions on differences between their handcrafted dialogue scenario and the one automatically generated by the system, to answer RQ2.

Phase 2

To answer RQ3, we conducted an interview study to collect qualitative insights into the usefulness of the authoring tool's learning support features. We asked an English as a

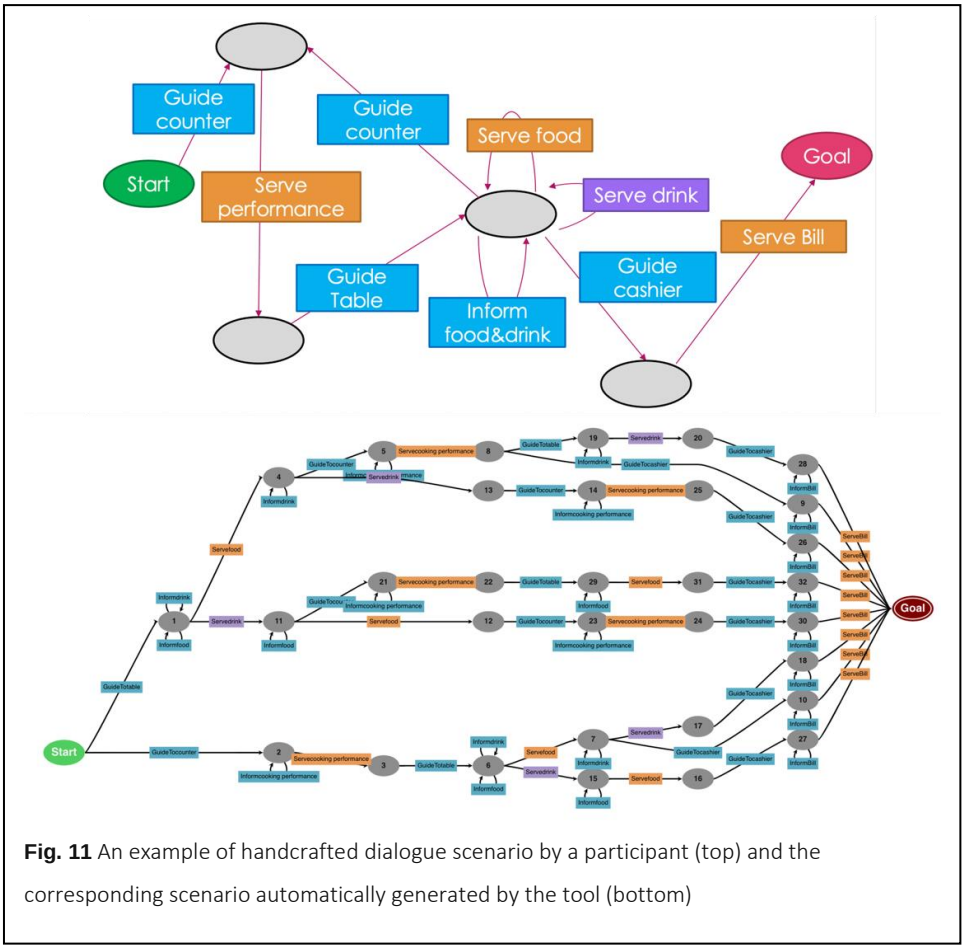

Fig. 11 An example of handcrafted dialogue scenario by a participant (top) and the corresponding scenario automatically generated by the tool (bottom)

Foreign Language (EFL) teaching expert (hereinafter called the expert) to evaluate the authoring tool's features from a language learning and teaching perspective. The expert has been teaching EFL in a technical senior high school for more than five years and is well-versed in the study's focus: communicative aspects of L2 acquisition. We provide the expert with an overview of the authoring tool and its features after explaining the motivation for our work. We also provided him with an extensive explanation of the authoring flow using concrete examples of several Services-oriented dialogue situations such as *Restaurant*, *Hotel Reception*, and *Gasoline Stand*. Then, we granted expert access to various dialogue scenarios created by the participants of Phase 1. The expert examined these dialogue scenarios and randomly selected and examined ten different dialogue scenarios (i.e., 5 handcrafted scenarios and corresponding 5 semi-automatically generated scenarios). We conducted an open-ended discussion with the expert after ensuring that he had enough time to examine the dialogue scenarios and understand the features implemented in the tool. We wanted to hear his opinions on the potential of the authoring interface for facilitating both L2 teaching and learning. Therefore, the following three central questions guided our discussion:

Question 1 (usefulness of the Scenario Type switching feature): it is possible to switch between basic and complex scenarios by using the scenario type property available in the dialogue scenario visualization interface. Do you think this feature can help with L2 acquisition in an EFL context? Please explain why you think so.

Question 2 (usefulness of the Payment Style switching feature): the tool also includes the ability to switch the payment style of a dialogue scenario from *PrePay* to *PostPay* and vice-versa. Do you think this feature has any potential for promoting L2 acquisition in the EFL context? Please explain why you think so.

Question 3 (overall impressions on the authoring tool's potential to foster second language acquisition): based on your answers to Questions 1 and 2, how would you evaluate the system's overall usefulness for language acquisition in the EFL context?

The interview was videotaped and the expert's responses to our questions were transcribed afterward. These transcripts were later shared with the expert, and he was allowed the opportunity to make clarifications and add his comments when necessary.

Results

Phase 1 results

In the following paragraphs, we report on participants' impressions on the usability and accessibility of the proposed authoring tool. We specifically report on differences observed in terms of perceived difficulty of the scenario generation task, and quality of dialogue

scenarios when comparing handcrafted dialogue scenarios to those automatically generated by the authoring tool.

Perceived difficulty of authoring task and amount of time spent on task: handcrafting vs automatic generation

As shown in Figure 12 (top), in terms of the level of perceived task difficulty when comparing the scenario handcrafting task to the generation via interface, most participants (83.3%) reported that they found the handcrafting task difficult (17 participants) or very difficult (3 participants). Several participants admitted that they found it quite hard to think of various dialogue paths when handcrafting the dialogue scenario on their own.

In terms of their perceptions of the amount of time spent on the task, most participants (75%) thought it was extremely long (4 participants) or long (14 participants), as shown in Figure 12 (bottom).

The amount of time spent on the handcrafting task also corroborated the above result. Participants took an average of 38 minutes (M = 37.74, SD = 5.22) to create their handcrafted dialogue scenarios after specifying the target dialogue scenario' constraints. Note that the dialogue task structure was specified beforehand (Step 2), and the authoring tool could instantly (less than 1 sec) generate the corresponding dialogue scenario.

Quality of dialogue scenarios: handcrafted scenarios vs automatically generated scenarios

Each participant was asked to compare the dialogue scenario generated by the system to their handcrafted scenario and report on the exhaustivity of the automatically generated scenario using a 5-point Likert scale ranging from *Exhaustive* to *Limited*. Most participants (80%) said the automatically generated scenario was exhaustive, whereas the remaining (20%) found it almost exhaustive.

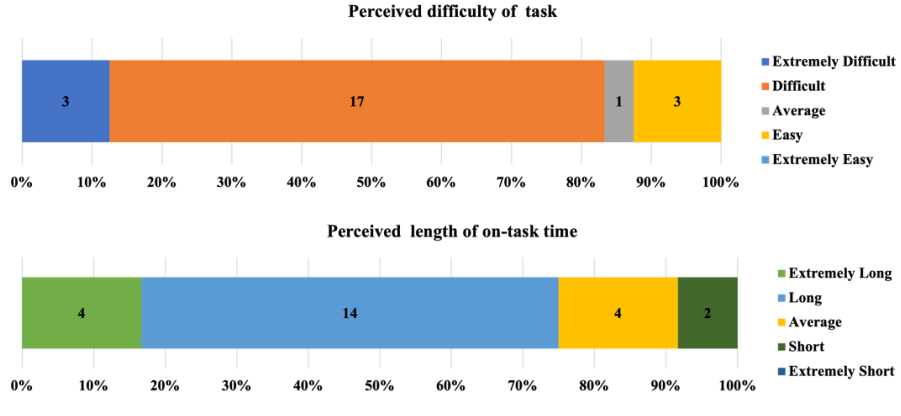

Fig. 12 Perceived difficulty and length of time spent on handcrafting of dialogue scenarios

Furthermore, to the Yes/No question "*In the automatically generated dialogue scenario, could you find any valid dialogue path that is missing in your handcrafted dialogue scenario?,*" the great majority (91.7%) of participants said that they noticed that the system could generate dialogue paths that were missing in their handcrafted dialogue scenarios. Note that the term valid here refers to dialogue paths that are aligned with defined constraints. Two participants reported the dialogue paths generated by the system were also covered in their handcrafted dialogue scenarios. However, a closer look at the dialogue scenarios designed by these two participants showed that both dialogue task structures contained only one instance of Core Action and one instance of Enhancing Action, making it relatively easy for them to handcraft all possible dialogue paths.

However, to the Yes/No question "*In your handcrafted dialogue scenario, is there any valid dialogue path you could not find in the dialogue scenario generated by the system?,*" 75% (18 out of 24) of participants reported failing to find any valid dialogue path defined in their handcrafted dialogue scenario but missing in the tool's scenario. Interestingly, the remaining 25% (6 out of 24) of participants stated that the system was unable to generate some of the dialogue paths that were defined in their handcrafted dialogue scenarios. Following a review of their handcrafted dialogue scenarios and follow-up interviews with these participants, two reasons were found for explaining this. First, some participants had defined in their handcrafted dialogue scenarios some "invalid" dialogue paths, such as cases where the same Core service Action appears more than once in the same dialogue path; obviously, the system could not have generated such paths. Second, others failed to notice that the dialogue paths they pointed out were generated by the tool. Therefore, the tool could generate all paths contained in participants' handcrafted dialogue scenarios.

In addition to these qualitative findings, we conducted a series of quantitative analyses to investigate the differences between both types of dialogue scenarios. Handcrafted dialogue scenarios contained an average of M = 3.25 (SD = 1.29) dialogue paths, whereas automatically generated dialogue scenarios had a numerically higher number of dialogue paths M = 6.33 (SD = 5.42). An independent samples *t*-test was used to test the hypothesis that the automatically generated dialogue scenarios and handcrafted dialogue scenarios contained a statistically different number of dialogue paths. The test was associated with a statistically significant effect ($t(46) = 2.71$, $p = .01$). Thus, the automatically generated dialogue scenarios contained a statistically significantly higher number of dialogue paths than the handcrafted ones. Cohen's *d* was estimated at 0.78, which is a large effect based on Cohen's (1992) guidelines.

Another independent sample *t*-test showed a significant difference ($t(46) = 3.99$, $p = .0001$) in the number of edges contained in handcrafted dialogue scenarios (M = 13.08, SD = 7.72), and automatically generated dialogue scenarios (M = 42.08, SD = 34.75). Hence, the automatically generated dialogue scenarios contained a statistically

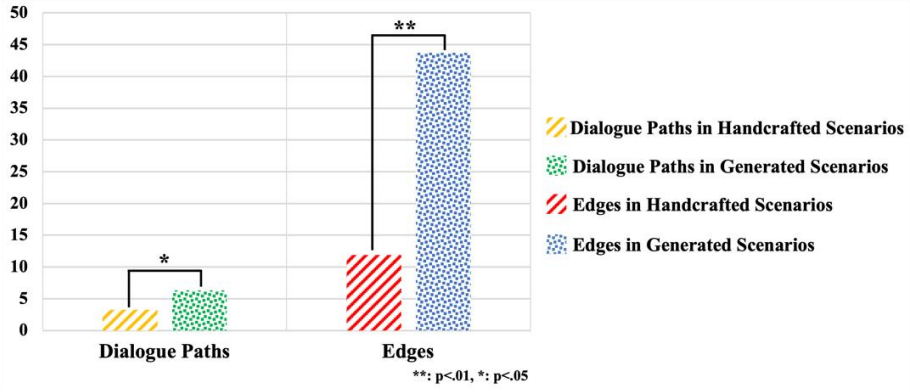

Fig. 13 Differences between handcrafted dialogue scenarios and automatically generated ones in terms of average number of dialogue paths and edges

significantly greater number of edges than the handcrafted ones. Cohen's *d* was estimated to be 1.15, indicating a significant effect based on Cohen's (1992) guidelines.

Figure 13 shows a graphical representation of these results.

Dialogue scenario preferences

To the extent of further evaluating participants' opinions on both dialogue scenarios, we asked them which dialogue scenario they would prefer to keep. 70% (17 out of 24) answered that they would prefer the automatically generated dialogue scenario, while 25% (5 out of 24) indicated that they would prefer their handcrafted scenarios. Two participants stated that they were unable to decide. Unsurprisingly, we discovered that all those who declared preferring their handcrafted dialogue scenarios were participants who stated incorrectly that they could design some valid dialogue paths, which were missing in the automatically generated scenarios, as previously explained.

Phase 2 results

The expert's answers to each of the topics discussed during the interview study are reported in the following paragraphs.

Expert's answer to Question 1 (usefulness of the Scenario Type switching feature)

"*Using this feature, one might be able to adjust the complexity of the dialogue scenario, which in turn may affect the difficulty level of the dialogue scenario. So, depending on whom (the teacher or the learner) is given the control of the scenario type switching, this feature may offer different possibilities for learning support. First, it makes sense that a teacher might want to set different difficulty levels to present learners with dialogue content that is adapted to their needs. On the other hand, this feature might also be helpful for*

fostering self-regulated learning since learners could use it to adjust the difficulty level of learning content by themselves. Besides, this could even be used by the dialogue system itself to dynamically adjust the level of presented dialogue contents without requiring human intervention thereby achieving a certain form of intelligent tutoring."

Expert's answer to Question 2 (usefulness of the Payment Style switching feature)

"*In terms of communication practice, this feature is interesting because without adding new content to the dialogue scenario, it modifies the scenario by reshaping the order in which dialogue contents are presented to learners. This could be interesting for prompting learners' use of different conversation strategies, when practicing similar dialogue contexts or contents multiple times. For this reason, this feature could be ideal to implement meaningful Pattern Practice, by enabling manipulation of the same linguistics resources, but in a less repetitive or predictable fashion.*"

Expert's answer to Question 3 (overall impressions on the authoring tool's potential to foster second language acquisition)

"*This authoring tool has the potential to open new perspectives in terms of L2 teaching and learning, because it offers the possibility to both teachers and learners to become actors in the design process of dialogue scenario contents. Besides, the tool enables easy generation of various patterns of the same scenario with varying levels of difficulty. From a learning support standpoint, if learners themselves are able to author or adjust the content of dialogue scenarios, they can generate their own preferred situations, which I think will increase their motivation to practice. On the other hand, from a teaching support perspective, this authoring interface can have very significant impact in the effective implementation of teaching methods such as Pattern Practice, which is widely used in the field of L2 acquisition.*"

Discussion and limitations

The above-described results allow us to draw several conclusions as far as each of our three research questions is concerned.

RQ1: Is the authoring interface user-friendly enough for human novices to overcome the difficulties associated with dialogue scenario specification?

How to simplify the use of authoring tools by non-programmer authors is crucial but unanswered or underexplored in current research (Dermeval et al., 2018). The results of our experiments indicate that there are several pieces of evidence to suggest that our proposed authoring tool may reduce the barrier to dialogue scenario authoring, allowing even for non-programmers to participate in the effective and rapid design of dialogue

scenarios. As shown in Figure 12, **the obtained results in terms of perceived difficulty and time on task indicate that our proposed tool may substantially reduce the authoring difficulty and time required to design service-related dialogue scenarios. First, in terms of authoring difficulty, our results showed that participants in our experiments found handcrafting dialogue scenarios more difficult than using the authoring tool. Similar conclusions can be drawn regarding authoring time, as the system outperformed human novices in terms of time required for scenario generation**.

As Murray (1999) mentioned, creating an explicit model of anything is not an easy task, and requires analysis, synthesis, and abstraction skills along with a healthy dose of creativity. The relatively important amount of time and cognitive resources required to keep track of the design progress, ensure that a given dialogue path is valid according to defined constraints, avoid designing redundant dialogue paths, and so on are all complicating factors for those who write dialogue scenarios in general. Our approach, which consisted in embedding a domain-independent generic model of Services task in the authoring tool allowed for the specification of the dialogue task based on only three categories of Service actions, which appears effective in reducing the difficulty of the dialogue scenario authoring task. Moreover, by allowing authoring via a user interface, we made it easier for people without programming knowledge to grasp the essence of the authoring procedure and intuitively use the tool. Furthermore, the dialogue task specification activity which was designed guild users through the authoring process step-by-step could have been quite beneficial in reducing the knowledge engineering effort involved in dialogue scenario design. Finally, scenario designers (i.e., experiment participants) may have found it easier to keep track of their progress through the design of dialogue scenarios by visualization of the task structure and the dialogue scenario.

RQ2: Are there any differences between scenarios generated automatically by the system and those designed manually by human novices?

Although authoring tools can significantly reduce the cognitive load involved in various design steps, it is difficult to reduce the entire design task to low-level decisions that result in a quality product. This is because high usability frequently comes at the expense of increased complexity and lower quality. Nevertheless, **our findings showed that, in addition to the proposed tool's accessibility and ease of use, the tool could generate dialogue scenarios that were of higher quality than human novices in most cases. The system-generated dialogue scenarios not only had more dialogue paths but also contained more edges than those handcrafted by human novices, as shown in Figure 13.** This suggests that our authoring tool could generate dialogue scenarios that were more exhaustive, and diverse than the handcrafted ones. This could explain why **most**

participants preferred the tool-generated dialogue scenario over their handcrafted ones. The high preference rate of dialogue scenarios generated by the system also indicates that the authoring tool went above the minimum needed to generate dialogue scenarios that could capture the designer's intentions. Let also mention that explanations provided by survey participants tend to confirm that the authoring tool was able to generate more valid and exhaustive dialogue scenarios. Especially among participants who defined multiple Core Actions and Enhancing Actions in their dialogue task structure, some praised the completeness of dialogue scenarios generated by the tool. In contrast, one might wonder why some participants, as reported in the results section, were unable to find particular differences between their handcrafted dialogue scenario and the automatically generated scenario. In this regard, note that only in cases when the dialogue task was not too complex (limited number of service actions), human novices could handcraft scenarios that were as exhaustive as the authoring tool's scenarios.

Based on the foregoing, we can reasonably conclude that the authoring tool presented in this study can generate dialogue scenarios at least as exhaustive as those designed by human novices and promote the good design of service-related dialogue scenarios.

RQ3: Do the proposed authoring interface and its features help promote second language learning and teaching?

Given that L2 learners' decision to initiate speech varies over time and across situations, a desirable conversational environment should provide such learners the opportunity to converse efficiently in various social conversation situations. Moreover, as suggested by the interaction hypothesis (Long, 1996), a well-designed interaction can help increase learners' engagement, lower their fear of the speaking task, and facilitate L2 acquisition. Therefore, providing educators with a tool to assist them in creating realistic dialogue scenarios could be a game-changer in L2 instruction. According to the EFL expert we interviewed, the inexpensive (i.e., does not require any additional authoring effort from the designer) one-click **dialogue scenario switching mechanisms implemented in our authoring tool may help educators provide EFL learners with tailored conversation opportunities considering learners' needs or proficiency. Furthermore, the expert also emphasized the tool's significant impact on the effective implementation of teaching methods such as Pattern Practice, which is widely used in the field of L2 acquisition. Moreover, since the tool is accessible to everyone, it could be used to directly involve learners in the design process of dialogue scenarios, increasing their autonomy, and putting them in control of their learning.** This will certainly offer new possibilities in terms of raising the potential of our authoring tool to fulfill other purposes such as fostering self-regulated learning process.

It is also worth noting that these findings further deserve credit in the sense that they hint at the feasibility of building authoring frameworks that could serve as a gateway for making accessible learning support system research outcomes to educators, ensuring that findings and innovations from research laboratories reach classrooms and actual learners.

In total, positive findings were obtained for all three of our research questions. Therefore, the aforementioned results can be viewed as a proof of concept, implying that it is possible to involve people who are unfamiliar with dialogue systems in the design process of intelligent conversational systems. In this light, the obtained results represent an important step toward providing L2 learners with computer-assisted realistic opportunities to simulate various conversation situations, practice their communicative skills, and reduce their apprehension about communicating in the target language.

Conversational systems that offer authentic interactions and simulation opportunities may be particularly beneficial for L2 learners in terms of improving both their cognitive and emotional readiness for communication. However, the relatively large number of skills and resources required for their implementation tempers the widespread adoption of such environments. Although some dialogue authoring frameworks have been developed in academia (Bohus & Rudnicky, 2009; Lison & Kennington, 2016), these tools were not designed to be accessible to people without programming skills and appear to suffer from a lack or low level of component reusability across various domains. Our findings provide evidence suggesting that these issues can be mitigated, making computer-supported conversational environments attractive to many educators. Our research findings further contribute to deepening the discussion on how to develop effective authoring tools relying both on human intelligence and artificial intelligence. According to Baker (2016), to achieve an intelligent design of learning support systems, one might need to develop authoring tools that take advantage of both machine and human intelligence in a balanced way. As suggested by Dermeval and colleagues, a natural way to accomplish this is to rely on non-programmer authors from the start of a learning support system (Dermeval et al., 2018). We think that the dialogue scenario authoring tool presented in this research provides a concrete example of how such a hybrid authoring of learning materials can be accomplished in the context of L2 acquisition.

Nevertheless, although our experiment has yielded some promising results, it is important to remember that building an authoring tool that is both easy to use and not too overly domain-specific remains extremely difficult. On a conceptual level, we acknowledge that additional work may be required to target dialogue situations that do not fall under the hood of service domains. Furthermore, we are aware that more work is still needed to confirm the effectiveness of the proposed tool. On one hand, the authoring tool must be tested by actual educators in real classroom settings to increase the quality of the user experience and measure actual learning gains through mid-long-term evaluations. Alternatively, we

should bear in mind that there is currently no agreement among researchers on what metrics should be considered when evaluating authoring tools for non-programmers authors, particularly, in terms of costs and usability. Therefore, we are unable to provide an objective discussion of how the findings reported here relate to previous studies dealing with similar issues at the time. Finally, note that aspects of the implementation of natural language processing and agent animation generation modules discussed in a previous contribution (Ayedoun et al., 2019), were omitted voluntarily from this paper to put more emphasis on the dialogue scenario authoring process itself.

Conclusion and future research directions

Task-oriented dialogue systems hold the promise of providing realistic computer-supported conversation opportunities for L2 learners to practice and acquire the target language. An essential component of dialogue systems is dialogue scenarios, which determine the conversation flow between the system and the users. In a language learning context, a well-designed dialogue scenario can help increase learners' engagement and thereby lower their fear of the speaking task. However, most teachers, educators, and people without related technical expertise find dialogue scenario design extremely difficult and inaccessible. In this study, we discussed the requirements that should ideally be fulfilled by a dialogue scenario authoring tool and indicated how these prerequisites can be addressed as a contribution to lowering the skill threshold associated with dialogue system design. The features of an authoring interface dedicated to facilitating the design of service-related dialogue scenarios in the context of L2 acquisition were then proposed and described.

We collected evidence of the tool's effectiveness in scaffolding the dialogue scenario design activity through a series of experimental evaluations. We observed that the proposed system could significantly reduce the perceived difficulty and time needed for generating dialogue scenarios. Furthermore, not only was the tool capable of generating qualitative dialogue scenarios, but we discovered that some features implemented in the tool could open up new opportunities for both L2 teachers and learners.

Future work will include the implementation of features toward further reducing dialogue task specification workload by making available more built-in ready-to-use dialogue components. We will also design features to facilitate the smooth and flexible integration of the dialogue scenario design module into other key modules of conversational agent modules such as CEWill. Finally, evaluation experiments will be conducted in real-world classroom settings to better understand the implications of using the authoring tool to support the tool's primary target users (i.e., educators and L2 learners). More evidence will also be collected to better understand how such an authoring tool may be used to impact

the second language acquisition. Last but not least, we will consider the feasibility of using the proposed tool to generate dialogue scenarios in domains other than service-related ones.

Abbreviations
EFL: English as a Foreign Language; L2: Second Language; WTC: Willingness To Communicate.

Acknowledgements
We would like to offer our special thanks to Mr. Keisuke Tanino for his meaningful comments during the interview study.

Authors' contributions
Conceptualization, Software, Writing-Original draft preparation and Funding acquisition: EA

Supervision, Review and Editing: YH

Methodology, Supervision and Validation: KS

All authors have read and agreed to the published version of the manuscript.

Authors' information
EA received his bachelor's, master's and Ph.D. degree in Informatics from Osaka Prefecture University, Japan, respectively in 2016, 2018 and 2020. From 2018 to 2020, he served as a young scientist research fellow (DC1) of the Japanese Society for the Promotion of Science (JSPS). He is currently Assistant Professor in the Faculty of Engineering Science, Kansai University. His research is at the intersection of Human-Computer Interaction, Affective Sciences and Educational Psychology. He is a member of the Japanese Society for Artificial Intelligence (JSAI), International Artificial Intelligence in Education Society (IAIED) and IEEE.

YH received his Ph.D. degree in Information Science from Nagoya University in 2012, and is currently an associate professor in the Graduate School of Informatics, Osaka Metropolitan University, Japan. His research interests include computer-supported collaborative learning and human-computer interaction. He received JSAI 30th Anniversary Best Paper Award in 2016, ICCE Best Overall Paper Award in 2017, and JSiSE Best Paper Award in 2021. He is a member of IPSJ, JSAI, JSiSE, HIS, and ACM.

KS received a Ph.D. from Osaka University in 1998, and is currently a professor in the Graduate School of Informatics, Osaka Metropolitan University, Japan. His research interests include intelligent tutoring systems, and ontological engineering. He received Best Paper Awards of the Japanese Society for Information and Systems in Education in 2012, 2015 and 2021 as well as the Best Overall Paper Award of the ICCE2017. He is a member of ACM and IAIED.

Funding
This research was supported by JSPS KAKENHI Grant Numbers #22K18011 and #18H03345.

Availability of data and materials
All of the data is contained within the article.

Declarations

Competing interests
The authors declare no conflict of interests.

Informed consent
Informed consent was obtained from all subjects involved in the study.

Author details
[1] Faculty of Engineering Science, Kansai University, 3-3-35 Yamatecho Suita, 564-8680 Osaka, Japan.

[2] Graduate School of Informatics, Osaka Metropolitan University, 1-1 Gakuencho Naka-ku Sakai, 599-8531 Osaka, Japan.

Received: 3 February 2022 Accepted: 17 October 2022
Published: 28 February 2023 (Online First: 28 December 2022)

References

Ayedoun, E., Hayashi, Y., & Seta, K. (2019). Adding communicative and affective strategies to an embodied conversational agent to enhance second language learners' willingness to communicate. *International Journal of Artificial Intelligence in Education, 29*(1), 29–57. https://doi.org/10.1007/s40593-018-0171-6

Ayedoun, E., Hayashi, Y., & Seta, K. (2020). Services task model based dialogue scenarios design towards L2 WTC support oriented dialogues authoring tool. In S. Yamamoto & H. Mori (Eds.), *Human Interface and the Management of Information. Interacting with Information. HCII 2020. Lecture Notes in Computer Science, vol 12185* (pp. 145–163). Springer, Cham. https://doi.org/10.1007/978-3-030-50017-7_10

Ayedoun, E., Hayashi, Y., & Seta, K. (2021). Authoring tool for semi-automatic generation of task-oriented dialogue scenarios. In M. M. T. Rodrigo et al. (Eds.), *Proceedings of the 29th International Conference on Computers in Education* (pp. 41–50). Asia-Pacific Society for Computers in Education.

Baker, R. S. (2016). Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, *26*(2), 600–614. https://doi.org/10.1007/s40593-016-0105-0

Bohus, D., & Rudnicky, A. I. (2009). The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language, 23*(3), 332–361. https://doi.org/10.1016/j.csl.2008.10.001

Cabada, R. Z., Estrada, M. L. B., & García, C. A. R. (2011). EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network. *Expert Systems with Applications*, *38*(8), 9522–9529. https://doi.org/10.1016/j.eswa.2011.01.145

Cohen, J. (1992). Statistical power analysis. *Current Directions in Psychological Science*, *1*(3), 98–101. https://doi.org/10.1111/1467-8721.ep10768783

Conlan, O., O'Keeffe, I., Brady, A., & Wade, V. (2007). Principles for designing activity-based personalized eLearning. In *Proceedings of Seventh IEEE International Conference on Advanced Learning Technologies* (pp. 642–644). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/icalt.2007.209

Dermeval, D., Paiva, R., Bittencourt, I. I., Vassileva, J., & Borges, D. (2018). Authoring tools for designing intelligent tutoring systems: A systematic review of the literature. *International Journal of Artificial Intelligence in Education*, *28*(3), 336–384. https://doi.org/10.1007/s40593-017-0157-9

du Boulay, B. (2016). Recent meta-reviews and meta-analyses of AIED systems. *International Journal of Artificial Intelligence in Education, 26*(1), 536–537. https://doi.org/10.1007/s40593-015-0060-1

Engström, H., Brusk, J., & Erlandsson, P. (2018). Prototyping tools for game writers. *The Computer Games Journal*, *7,* 153–172. https://doi.org/10.1007/s40869-018-0062-y

Farkhadov, M., Petukhova, N., Eliseev, A., & Farkhadova, M. (2019). How to design dialogue scenarios and estimate main dialogue parameters for a voice-controlled man-machine interface. In *Proceedings of 2019 IEEE 13th International Conference on Application of Information and Communication Technologies* (pp. 1–5). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/AICT47866.2019.8981770

Ferrario, R., & Guarino, N. (2008). Towards an ontological foundation for services science. *Future Internet Symposium*, 152–169. https://doi.org/10.1007/978-3-642-00985-3_13

Ferrario, R., Guarino, N., Janiesch, C., Kiemes, T., Oberle, D., & Probst, F. (2011). Toward an ontological foundation of services science: The general service model. In *10th International Conference on Wirtschaftsinformatik, 2*, 675–684.

Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education, 48*(4), 612–618. https://doi.org/10.1109/te.2005.856149

Johnson, W. L. (2019). Data-driven development and evaluation of Enskill English. *International Journal of Artificial Intelligence in Education*, *29*(3), 425–457. https://doi.org/10.1007/s40593-019-00182-2

Kerr, P. (2016). Personalization of language learning through adaptive technology: Part of the *Cambridge Papers in ELT series*. Cambridge: Cambridge University Press.

Lester, L., Mott, B., Rowe, J., & Taylor, R. (2015). Design principles for pedagogical agent authoring tools. In R. Sottilare, A. Graesser, X. Hu & K. Brawner (Eds.), *Design recommendations for intelligent tutoring systems: Volume 3 - Authoring tools and expert modeling techniques*. U.S. Army Research Laboratory.

Lison, P., & Kennington, C. (2016). OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *Proceedings of ACL-2016 System Demonstrations* (pp. 67–72). Association for Computational Linguistics. https://doi.org/10.18653/v1/p16-4012

Long, M. H. (1996). The role of the linguistic environment in second language acquisition. In W. R. Ritchie & T. Bhatia (Eds.), *Handbook of second language acquisition* (pp. 413–468). Academic Press. https://doi.org/10.1016/b978-012589042-7/50015-3

Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology, 106*(4), 901–918. https://doi.org/10.1037/a0037123

MacIntyre, P. D., Clement, R., Dörnyei, C. Z., & Noels, K. A. (1998). Conceptualizing willingness to communicate in a L2: A situational model of L2 confidence and affiliation. *Modern Language Journal*, *82*(4), 545–562. https://doi.org/10.1111/j.1540-4781.1998.tb05543.x

Matsuda, N., Cohen, W. W., & Koedinger, K. R. (2015). Teaching the teacher: Tutoring SimStudent leads to more effective Cognitive Tutor authoring. *International Journal of Artificial Intelligence in Education, 25*(1), 1–34. https://doi.org/10.1007/s40593-014-0020-1

Meron, J., Valente, A., & Johnson, W. L. (2007). Improving the authoring of foreign language interactive lessons in the Tactical Language Training System. In *Workshop on Speech and Language Technology in Education*, 33–36. https://doi.org/10.1609/aimag.v30i2.2240

Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., & Mcguigan, N. (2009). ASPIRE: An authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 155–188. https://doi.org/10.1007/11774303_5

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education, 10*, 98–129. https://doi.org/10.1007/978-94-017-0819-7_17

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education, 24*(4), 427–469. https://doi.org/10.1007/s40593-014-0029-5

Paquette, L., Lebeau, J. F., Beaulieu, G., & Mayers, A. (2015). Designing a knowledge representation approach for the generation of pedagogical interventions by MTTs. *International Journal of Artificial Intelligence in Education, 25*(1), 118–156. https://doi.org/10.1007/s40593-014-0030-z

Preuss, S., Garc, D., & Boullosa, J. (2010). AutoLearn's authoring tool: A piece of cake for teachers. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 19–27). Association for Computational Linguistics.

Raux, A., & Eskenazi, M. (2004). Using task-oriented spoken dialogue systems for language learning: Potential, practical applications and challenges. In *InSTIL/ICALL Symposium*.

Reinders, H., & Wattana, S. (2014). Can I say something? The effects of digital game play on willingness to communicate. *Language Learning & Technology, 18*(2), 101–123.

Susarla, S., Adcock, A., Van Eck, R., Moreno, K., Graesser, A. C., & The Tutoring Research Group. (2003). Development and evaluation of a lesson authoring tool for AutoTutor. In *AIED2003 Supplemental Proceedings*, 378–387.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197–221. https://doi.org/10.1080/00461520.2011.611369

Woolf, B. P. (2010). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-373594-2.X0001-9

Publisher's Note