Rodrigo, M. M. T. et al. (Eds.) (2021). Proceedings of the 29th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Authoring Tool for Semi-automatic Generation of Task-Oriented Dialogue Scenarios

Emmanuel AYEDOUN[a*], Yuki HAYASHI[b] & Kazuhisa SETA[b]
[a]*Faculty of Engineering Science, Kansai University, Japan*
[b]*Graduate School of Humanities and Sustainable System Sciences, Osaka Prefecture University, Japan*
*emay@kansai-u.ac.jp

Abstract: The lack of suitable conversation opportunities is often pointed out as a major factor inhibiting second language learners' willingness to communicate in the target language. Although computer-based conversational environments have been advocated as a promising approach to mitigate this issue, high authoring costs still prevent their widespread adoption. In this paper, we present a dialogue scenario authoring system that could facilitate the rapid implementation of desirable situational dialogue scenarios, thereby lowering the dialogue scenario authoring barrier for non-programmers or even educators. To this extent, we exploit the common underlying structure of services (restaurant, hotel, travel-planning, etc.) that seem to share a certain degree of similarity at the task slevel and built a versatile dialogue scenario authoring interface that enables semi-automatic generation of services-related dialogue scenarios. Here, we describe the features of the proposed system, and present the results of a pilot evaluation study that hint on the meaningfulness of our approach towards facilitating dialogue scenarios authoring by people who do not have any previous experience designing dialogue systems components.

Keywords: Authoring tools, dialogue scenario design, adaptive language learning, conversational agents, willingness to communicate in L2

1. Introduction

In the field of second language (L2) acquisition, researchers have reported that many learners feel genuine anxiety about performing in front of others, concluding that many classrooms do not offer learners much in the way of communicative practice (Reinders, & Wattana, 2014). On the other hand, it was reported that computer-supported dialogue environments could be effective towards providing L2 learners with realistic opportunities to simulate daily conversations, whereby to alleviate emotional variables that inhibit learners' motivation towards communication in L2.

For instance, we developed an embodied conversational agent (CEWill) that provides second language learners with task-oriented spoken dialogue simulation opportunities in a restaurant context (Ayedoun, Hayashi, & Seta, 2019). Dialogue scenarios in CEWill were designed following a knowledge-based approach enabling it to achieve a deeper level of understanding and control of the conversation flow, increasing the degree of reality of interactions. However, the significant level of knowledge engineering effort and the degree of dialogue expertise that is necessary for implementing new scenarios in this system constitute an important obstacle that may limit its adoption and frequent use in real educational settings.

To alleviate such issues and promote the availability of a rich pool of realistic dialogue scenarios for second language learners, our goal is to build a dialogue scenario authoring tool that could facilitate the rapid implementation of desirable dialogue scenarios and lowering the authoring barrier for non-programmers or educators, who are not necessarily knowledge or software engineers. To this extent, we exploit the common underlying structure of services (restaurant, hotel, travel-planning, etc.) that seem to share a certain degree of similarity at the task level, and build a dialogue scenario authoring interface that enables semi-automatic generation of dialogue scenarios across various services domains.

In the present study, we present the developed dialogue scenario authoring tool, describe some of its features, and report on the meaningfulness of our approach towards reducing the authoring barrier for people who are not necessarily familiar with dialogue systems or dialogue scenario design.

2. Related Works

2.1 Authoring Tools and Learning Support

Several recent reviews have noted the effectiveness of learning support systems and particularly intelligent tutoring systems, highlighting that, well-designed systems can successfully complement or substitute other instructional models in many common academic subjects (du Boulay, 2016; Ma, Adesope, Nesbit, & Liu, 2014; Van Lehn, 2011). However, these tutoring systems remain hard to author. 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, Martin, Suraweera, Zakharov, Milik, Holland, & Mcguigan, 2009), ASTUS (Paquette, Lebeau, Beaulieu, & Mayers, 2015), AutoTutor tools (Nye, Graesser, & Hu, 2014), SimStudent (Matsuda, Cohen, & Koedinger, 2015) have been proposed and most of them do not require advanced programming.

Authoring tools for conversation-based learning environments have focused on assisting non-technical users in the creation of pedagogical agent dialogues. AutoTutor (Graesser, Chipman, Haynes, & Olney, 2005) provides multi-agent conversational interactions to tutor students using the discourse patterns of a human tutor, and has been used across multiple domains including computer literacy and physics. To facilitate the application of AutoTutor to other domains, authoring tools have been developed to aid subject matter experts in creating dialogue-based tutors, such as the AutoTutor Script Authoring (Susarla, Adcock, Van Eck, Moreno, & Graesser, 2003) and AutoLearn (Preuss, Garc, Boullosa, 2010). Similarly, an authoring tool has been created for the Tactical Language and Culture Training System (TLCTS) that allows subject matter experts to create pedagogical dialogue for a foreign language learning training system at reduced costs (Meron, Valente, & Johnson, 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, Mott, Rowe, & Taylor, 2015). The available authoring tools environments, although certainly useful to implement pedagogical agents for specific domains, still seem to suffer from a lack or limited level of abstraction or versatility of their encapsulated initial domain knowledge, which limits the reusability of their key components across different domains.

2.2 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, O'Keeffe, Brady, & Wade, 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, we proposed CEWill, an embodied conversational agent that provides second language learners with opportunities to freely simulate spoken dialogue in realistic daily-life settings such as talking with a waiter in a restaurant (Ayedoun, Hayashi, and Seta, 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. Results of experimental evaluations provided insights on the meaningfulness of such simulation environment, especially in countries where English learning focuses

Figure 1. CEWill interface and learner interacting with the agent Peter in restaurant scenario (Ayedoun, Hayashi, and Seta, 2019).

less on the development of communicative skills and where learners have limited access to opportunities for using the target language in authentic settings.

However, to achieve a high degree of reality in interactions similar to what learners are likely to experience in daily face-to-face situations, we have also suggested that a careful design of the different dialogue scenarios would be paramount. 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 could potentially constitute a barrier that may prevent the large diffusion and adoption of dialogue simulation environments such as CEWill.

3. Research Objective and Requirements

3.1 Pertinence of Issue and Research Objective

Conversational systems that provide authentic interactions simulation opportunities might be particularly beneficial for second language learners in terms of enhancing both their cognitive and emotional readiness towards communication. Yet, the widespread diffusion of such environments is tempered with the relatively important number of skills and resources required for their implementation. To address such an issue, some dialogue authoring frameworks have been developed in academia (Bohus, & Rudnicky, 2009; Lison, & Kennington, 2016). However, designing dialogue scenarios for various situations using these tools remains a challenge due to a lack or low level of reusability of their components across various domains. Murray (1999) hinted at the challenging trade-off issue related to the extent to which 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, but general enough to be attractive to many educators.

In the light of the above, the present study aims to propose a flexible authoring environment that could ease the design of dialogue scenarios over a relatively wide range of different conversation situations. If achieved, such balance between ease of authoring and flexibility of the resulting tool will allow scenario authors (i.e., educators) to have better control over the specification of dialogue behaviors, which is a crucial requirement for conversational applications, especially in the field of education. Doing so would also ultimately promote the availability of a rich pool of dialogue scenarios for L2 learners, which is desirable in terms of enhancing their motivation towards communication.

To this extent, we aim 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 domain-independent reusable dialogue task components. Doing so will facilitate the design process of various dialogue scenarios falling under the hood of the targeted common task. 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.

3.2 Requirements and Fulfillment

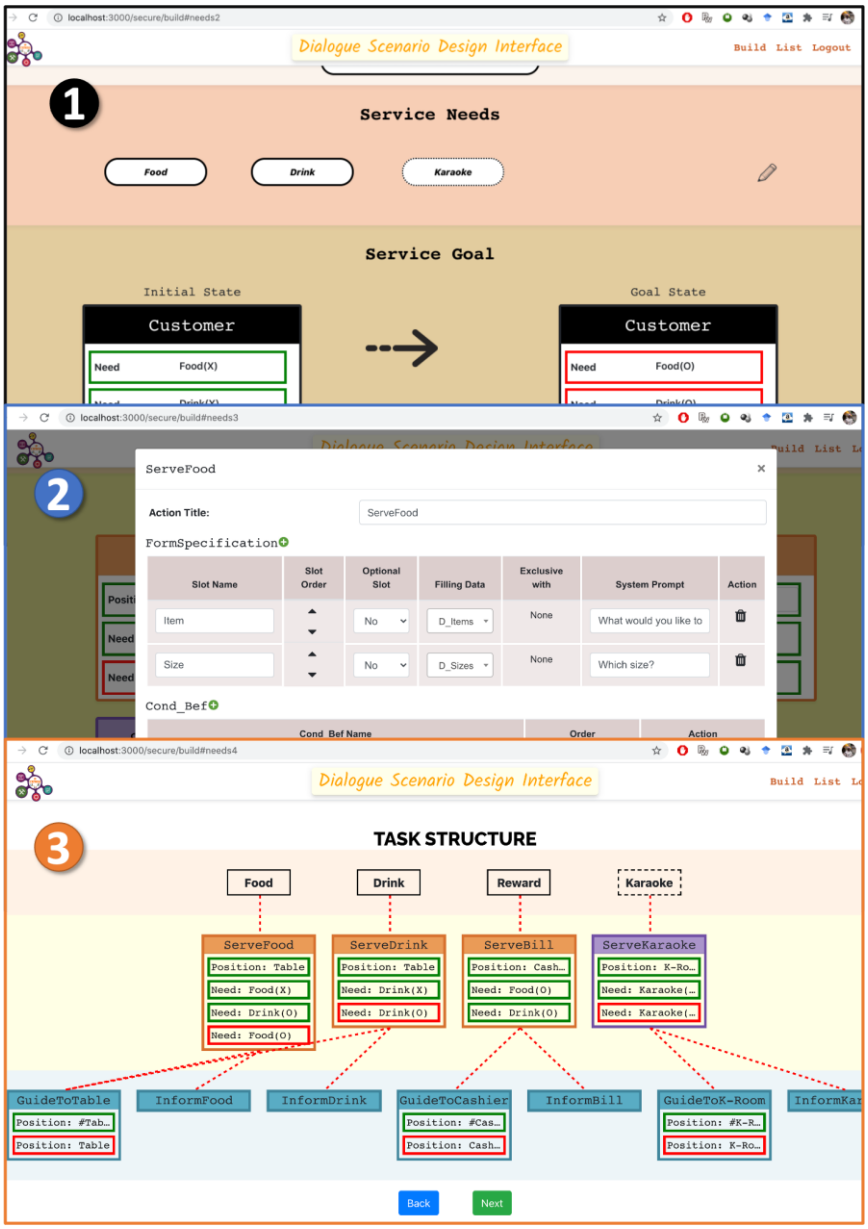
Achieving a good level of versatility while still maintaining the authoring costs at an acceptable level for content creators is a challenging task, but certainly necessary to make possible a more active and

frequent use of authoring tools and learning support systems by educators who are not necessarily software engineers. Keeping this in mind and inspired by Murray (1999)'s review on authoring tools, we have identified several core requirements for the desired dialogue scenario authoring tool in the context of this study. In the following lines, we describe these prerequisites and explain how we address them in the built tool.

- **Embed a relevant level of domain-independent knowledge about task structure:** this refers to some generic knowledge about the common structure of the different dialogue domains to be targeted by the system. Such knowledge, if pre-wired and embedded in the tool, could make authoring easier and more powerful through the reuse of the same structure across various instances of dialogue domains. In such a way, dialogue scenario authors could just focus on specifying domain-specific aspects of the dialogue flow, which will significantly lessen new scenarios implementation effort and contribute to decreasing authoring time and costs. Our key idea towards covering a wide range of different dialogue domains (e.g., restaurant, hotel, transportation) is to embed a reusable domain-independent task model (services task model) in the authoring tool, as proposed in our previous work (Ayedoun, Hayashi, & Seta, 2020). Doing so could help developers with no or limited programming skills, to design and implement new service domain-related conversation scenarios in a cost-effective fashion, mostly through the specification of properties for key task components in the target domain.
- **Make possible efficient authoring flow and knowledge management:** this involves the system's ability to scaffold the dialogue scenario specification by allowing input through the use of templates, data entry forms, pop-up menus, etc. In our proposed authoring tool, whenever the range of possible input values can be limited to a finite set, scenario authors are allowed to select rather than type. Besides, we made clear separation about the different types of information that scenario authors have to deal with (i.e., actions specifications, interface parameters) and those that the system takes care of discretely behind the scene (goals and actions structure generation, dialogue flow generation, interface parameters) as we will explain in the following section. This is expected to contribute to decreasing the skill threshold for dialogue scenario modeling and allow actual educators and other people with non or less programming skills to take part in the dialogue scenario specification process.
- **Enable scenario authors to apprehend the structure of authored dialogue scenarios:** our proposed authoring tool features a user-friendly interface that allows authors to see both the static structure of the designed dialogue scenario (later shown in Figure 2(3)) and the dialogue control dynamics over possible dialogue paths (later shown in Figure 3). This enables scenario authors to interactively participate in the authoring process, examine the authored dialogue structure and make necessary refinements to achieve the desired dialogue scenario. This might especially be desirable for reducing the cognitive load associated with the design of complex dialogue scenarios.
- **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 shared dialogue scenarios design principles 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 domain-specific dialogue flow logic. As far as this point is concerned, the dialogue scenario built with the present authoring tool is intended to be fully compatible with conversational agent systems such as CEWill so that there is no need to implement from scratch other key components dedicated to natural language processing and animation generation. Aspects related to natural language recognition and generation are thus voluntarily omitted from this paper to put more emphasis on the dialogue scenario authoring process itself.

4. Authoring Dialogue Scenarios

Figure 2 shows different windows of the built authoring tool. The system is accessible via browser and comprises the following four different windows corresponding to each stage of the authoring process: the dialogue task specification window (Figure 2(1)), the slot customization window (Figure 2(2)), the dialogue task structure visualization window (Figure 2(3)), and finally the dialogue scenario visualization window shown in Figure 3.



Fiaure 2. Authoring Tool Interface showing Various Windows.

In order to ease the authoring process, the system embeds a generic model of services accordingly to the conceptual framework proposed in (Ayedoun, Hayashi, & Seta, 2020), and inspired by Ferrario's prior work (Ferrario, Guarino, Janiesch, Kiemes, Oberle, & Probst, 2011). According to this framework, any service-oriented dialogue scenario can be expressed as a combination of the following three key components:

- **Core service Actions:** are those actions whose execution contributes to satisfying users' needs. In da sense, these actions characterize a service for what it is and must necessarily be exposed to the customer, e.g., for restaurant service, the action of *serving foods* or *serving drinks*;
- **Supporting Actions:** are actions necessary to the service but not explicitly mentioned as constituting the service, e.g., for restaurant service, the action *of guiding the customer to a seat* or *explaining the menu*;
- **Enhancing Actions:** are actions meant to augment the value of the service. These actions can be considered as additional services actions that are connected to but not strictly included in the main service, e.g., for restaurant service, the action of *offering karaoke*, or any other *entertainment as option*.

Along the authoring process, using the above service actions, the scenario designer specifies a hierarchically-organized service task which is similar to a type of plan structure defined in AI planning, for covering a topic. Each dialogue scenario addresses high-level goals (i.e., Needs to be satisfied) and

is generated by the system following the designer's specifications as a sequence of any combination of core service actions, supporting actions, and enhancing actions. Note that the scenario designer is just required to specify execution constraints for each service action without having to care about the execution flow (i.e., how actions combine with one another), which is rather handled behind the scenes by the system. 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 domain dialogue scenario, while the authoring tool exploits the common underlying structure (i.e., service model) to manage inter-domain commonalties.

From an L2 learning support perspective, designing the dialogue scenario as a combination of these three types of service actions provides much flexibility in terms of generating dialogue content that is personalized to the needs of learners. For instance, dialogue scenarios including only core service actions (i.e., basic scenarios) may be used for beginners, while scenarios including also enhancing actions (i.e., extended scenarios) may be presented to advanced learners. That is, the dialogue system will be able to dynamically adapt the scenario content to learners' level without requiring any additional content authoring effort from scenario designers.

4.1 Dialogue Task Specification at Macro-level

The first step in the authoring process consists in specifying key components (i.e., service actions) of the targeted dialogue task structure. To begin with, the dialogue scenario author inputs customer's Need(s) to be satisfied by the target service, as shown in Figure 2(1). For example, in a restaurant scenario, *Drink* and *Food* may be set as primary Needs. According to these Need(s), the tool generates the service goal which is a representation of the initial state and goal state 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 author may add a Spatio-temporal requirement (i.e., Position) for the service delivery, which can take different values at initial (e.g.: *Entrance*) and goal (e.g.: *Cashier*) states. This refinement of the service Goal enables scenario authors to clarify the big picture of the target service delivery process. At this point, the tool becomes able to generate base structures of Core and Enhancing actions to be executed to satisfy Need(s) specified in the service Goal. The suitable types of actions are automatically set according to the nature of the target Need(s). Based on the automatically generated basic structure of Core and Enhancing Actions, the scenario author may set additional Spatio-temporal requirements or constraints for each action, if necessary. Moreover, basic structures of Supporting actions are automatically generated and attached to each defined Core action and Enhancing action, as can be seen in Figure 2(3).

4.2 Dialogue Task Specification at Micro-level

Each service action may have a certain number of slots. In terms of actual dialogue flow, note that 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 dialogue agent and the learner. Although default slots are already predefined for convenience, the scenario author can

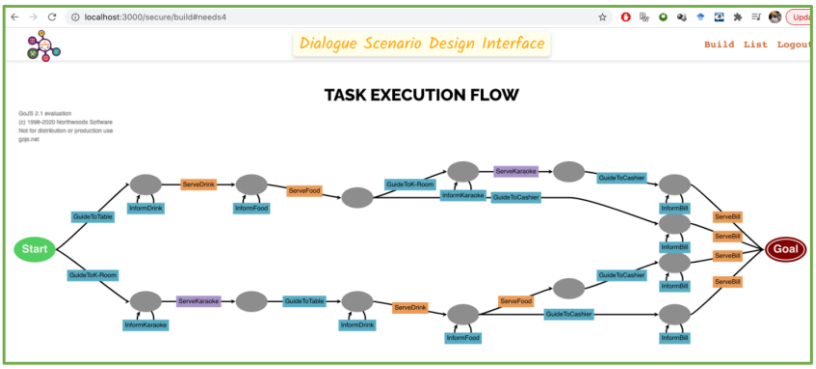


Figure 3. Authoring Tool Interface showing Automatically generated Dialogue Scenario.

still add new ones or customize exiting ones by specifying several properties that will determine the flow of slot-filling during actual dialogue. To ease the handling of this essential activity, the scenario author is prompted with a slot specification window, as illustrated in Figure 2(2).
Slots properties that are customizable include:

- **Order property:** indicates the order in which slots need to be filled to make the dialogue sound more natural. For instance, as far as the example presented in Figure 2(2) is concerned, it might be reasonable to have the slot *Item* get filled prior to the slot *Size* since different *Size* (e.g., Shot, Medium, Pitcher, Glass, Bottle) options might be available or not, depending on the type of *Item* (e.g., Beer, Wine, Tequila) that is selected.
- **Optional property:** indicates whether the slot's value is indispensable or not for the target service action execution.
- **Filling data property:** constrains the semantic type of the target slot. The configuration is conducted by selecting the appropriate type among the predefined ones. Scenario authors can still customize the existing slot types according to the restrictions of the target domain, or define new types from scratch.
- **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 redundancy in slot-filling.
- **System prompt property:** shows the system prompt for triggering the learner's answer and fill the target slot. For example, when the target slot is Item under the core-service action *ServeDrink*, the system prompt might be specified as "*What would you like to drink?*" or "*Anything to drink?*".

4.3 Dialogue Task Structure Examination

The authored dialogue task structure for the target dialogue scenario is displayed as a hierarchical structure allowing the scenario author to debug and grasp the big picture of the authored domain knowledge. As illustrated in Figure 2(3), this can be seen as a static task representation of the dialogue scenario. Relations between customer Needs and service actions can be revisited by the scenario author, to ensure that intended service specification is achieved. For instance, the scenario author may decide to make appropriate revisions by adding missing constraints, or even defining new customer Needs, if necessary.

4.4 Dialogue Scenario Examination

The resulting dialogue scenario or task execution flow is automatically generated by the system, as shown in Figure 3. This can be seen as an abstract representation of the dialogue flow showing the collection of all possible task execution paths or dialogue paths with respect to constraints specified by the scenario designer through the dialogue task specification activity. The dialogue scenario is outputted by the tool in the form of a finite state machine, where each state (i.e., node) corresponds to any milestone somewhere between the initial state and the goal state; arcs (i.e., edges) connecting different states stand for service actions. Using the information displayed on this window, the scenario author can visually apprehend in which order each service action might be executed as dialogue unfold between the agent and the learner. Undesirable dialogue paths may be removed, and missing ones can be added by revisiting the specifications in the task specification windows (Figure 2(1) and 2(2)). At this point, if the author is satisfied with the generated dialogue scenario, the whole specification data can be saved for further editing or exported for integration into a conversational agent environment such as CEWill.

In the example shown in Figure 3, one can notice that the system was able to generate several dialogue scenarios for the same domain (i.e., restaurant). For instance, dialogue scenarios involving only core actions (i.e., *ServeFood*, *ServeDrink*) and corresponding supporting actions (e.g., *InformDrink*, *GuideTable*) were generated along with more extensive dialogue scenarios that also included an enhancing action (i.e., *ServeKaraoke*).

5. Pilot Study and Results

5.1 Research Questions and Experimental Settings

We conducted an experimental study to evaluate the usability of the authoring tool as well as its effectiveness in terms of reducing authoring time and preserving quality of generated dialogue scenarios. In short, the design of this preliminary experiment was guided by the necessity to investigate the following three research questions:

RQ1: Is the authoring tool accessible enough for human novices in terms of alleviating necessary knowledge engineering effort needed for dialogue scenario specification?

RQ2: Is there any noticeable difference between amount of cost needed by the system for dialogue scenarios generation compared with cost required by human novices for manual generation?

RQ3: How exhaustive are the scenarios semi-automatically generated by the system compared to those manually designed by human novices?

To answer these research questions, we designed an experimental setting to evaluate the system usability (RQ1), compare dialogue scenarios designed by human novices to those generated by the system, and see whether we could find any differences in terms of authoring time (RQ2) and quality (RQ3) of both types of dialogue scenarios. Note that the term "human novices" is used here to describe people who are not familiar with dialogue scenario design or do not have any technical experience in terms of designing dialogue systems.

The system was tested by seven undergraduate and graduate students in engineering with no previous experience authoring dialogue scenarios. We provided the participants with a short set of training materials and training tasks, which we guided them through. This required about 25 minutes. We then let them use the authoring tool to freely author the dialogue scenario they wish to implement. To ensure task homogeneity among participants, we advise them to think of dialogue in a restaurant context since it is a dialogue domain that even human novices should be reasonably familiar with.

After allowing participants to freely specify the key components of their dialogue task structure (contents of sections 4.1 to 4.3) using the tool, we asked them to stop the authoring process one step before checking the generated dialogue scenario (section 4.4). Then, at this point, to elucidate RQ2, we asked them to hand-design a dialogue scenario that would satisfy the constraints of their dialogue task structure using finite state machine symbols. We gave them as much time they needed and allowed them to check the task structure they priorly built on the interface if necessary. After they were done with this, we allow them to look at the dialogue scenario generated by the system. Then, we administrated a survey questioning their perceived difficulty of the dialogue scenario hand-design task, to elucidate RQ1, as well as their opinions on differences between their dialogue scenario and the one generated by the system, in order to investigate RQ3.

5.2 Results

RQ1: Is the authoring tool accessible enough for human novices in terms of alleviating necessary knowledge engineering effort needed for dialogue scenario specification?

In terms of the level of perceived task difficulty when designing dialogue scenarios, most participants (6 out of 7) reported that they found it difficult or very difficult to think of several dialogue paths when designing the dialogue scenario on their own (i.e., manually). On the other hand, they found that the authoring task was easier when using the authoring tool.

RQ2: Is there any noticeable difference between amount of cost needed by the system for dialogue scenarios generation compared with cost required by human novices for manual generation?

The above result was also corroborated by the amount of time spent on task. After specifying the target dialogue scenario' constraints, participants took on average about 35 minutes (M= 35.14, SD: 4.22) to come up with relatively simple dialogue scenarios. Note that since the scenario constraints were defined beforehand, the authoring tool was able to instantly (less than 1 second) generate the corresponding dialogue scenario. In other words, the system was in average, roughly more than 2100 (35 min * 60 sec) times, faster than human novices in outputting the corresponding dialogue scenario.

RQ3: How exhaustive are the scenarios semi-automatically generated by the system compared to those manually designed by human novices?

As far as differences between both dialogue scenarios are concerned, we found that valid dialogue scenarios generated by the system contained on average about 29% more edges (M=29.06, SD=8.60) than the participants' ones. This suggests that the tool was in most cases, able to generate more exhaustive dialogue scenarios than the participants. This was further corroborated by the survey results, as most participants (5 out of 7) found the dialogue scenario generated by the system completer and more exhaustive than their own. Some even mentioned that the system's dialogue scenario included some dialogue paths they have not thought of beforehand. Interestingly, two participants reported that they could not find any particular difference between theirs and the system's ones.

5.3 Discussion and Limitations

The results reported above seem to suggest that our proposed tool may substantially decrease the authoring difficulty (RQ1) involved in designing service-related dialogue scenarios. Provided that constraints on the dialogue task structure have been well-specified, even in the worst case, the system was able to generate dialogue scenarios that are at least, as exhaustive (RQ3) as those designed by human novices. In addition, the system obviously outperformed human novices in terms of time required for scenario generation (RQ2). It follows that, positive insights were obtained for each of our three research questions stated earlier in section 5.1. From these results, we hypothesise that the dialogue task specification activity which was designed for guiding the tool users step-by-step through the authoring process might have been quite beneficial towards reducing the knowledge engineering effort involved in dialogue scenario design. In addition, through the task structure and the dialogue scenario examination, experiment participants may have found it easier to keep track of their work.

These premises can be viewed as a proof of concept, suggesting that it is feasible to allow people who are not even familiar with dialogue systems, such as second language educators for example, to get involved in the design process of intelligent conversational systems. Seen in this perspective, obtained results are an important milestone towards providing second language learners computer-supported realistic opportunities to simulate various conversation situations, practice their communicative skills and reduce their apprehension towards communication in the target language.

It also seems important to emphasize that these results further deserve credit in the sense that they hint on the feasibility of building authoring frameworks that could serve as a gateway for making accessible learning support system research' outcomes to educators, so that findings and innovations from research laboratories actually reach classrooms and actual learners.

Nevertheless, although our experiment has produced some promising results, we are aware that more work is still needed to confirm the effectiveness of the proposed tool. On the one hand, the authoring tool has to be tested by actual educators to increase the quality of the user experience with its interface and validate its effectiveness. On a more conceptual level, we acknowledge that further work might be required for targeting dialogue situations which do not fall under the hood of service domains.

6. Conclusion and Future Research Directions

Authoring tools are necessary to support the more rapid delivery of computer-based learning supporting systems. However, building an authoring tool that is easy to use and not too domain specific remains extremely difficult. In this paper, we discussed on requirements that should ideally be fulfilled by a suitable authoring tool, and indicated how these prerequisites can be addressed. Then, we presented an authoring interface that enables semi-automatic generation of services related task-oriented dialogue scenarios for the field of second language learning. An experimental evaluation suggested that the proposed system could lower the dialogue scenario authoring barrier for people with no prior experience with dialogue systems, hinting on the meaningfulness of our approach.

Directions for future works include the implementation of features towards further reducing dialogue task specification workload by making available more built-in ready-to-use dialogue components. We will also devise features towards facilitating smooth and flexible integration of the dialogue scenario design module into other key modules of conversational agents such as CEWill. Finally, evaluation experiments will be conducted in real classroom settings to further understand the implications of using the authoring tool to support the tool's target (i.e., educators). More evidence will

also be collected to understand how such authoring tool may impact students' learning in general and second language acquisition in particular.

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.

Ayedoun, E., Hayashi, Y., & Seta, K. (2020). Services Task Model Based Dialogue Scenarios Design Towards L2 WTC Support Oriented Dialogues Authoring Tool. *In International Conference on Human-Computer Interaction*, 145-163.

Bohus, D., & Rudnicky, A. I. (2009). The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language, 23*(3), 332-361.

Conlan, O., O'Keeffe, I., Brady, A., & Wade, V. (2007). Principles for designing activity-based personalized eLearning. In *IEEE International Conference on Advanced Learning Technologies*, 642-644.

du Boulay, B. (2016). Recent meta-reviews and meta-analyses of AIED systems. *International Journal of Artificial Intelligence in Education, 26*(1), 536-537.

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.

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, and K. Brawner (Eds.) *Design Recommendations for Intelligent Tutoring Systems: Volume 3 - Authoring Tools and Expert Modeling Techniques*, Orlando, FL: U.S.

Lison, P., & Kennington, C. (2016). OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *Proceedings of ACL-2016 System Demonstrations*, 67-72.

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.

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.

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.

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.

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.

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.

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.

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*, 19-27.

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.