Goal-Oriented Modelling for Virtual Assistants

Jonathan Leung*†, Zhiqi Shen*, Chunyan Miao*
*School of Computer Science and Engineering
*Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly
†Joint NTU-WeBank Research Centre on Fintech
Nanyang Technological University
50 Nanyang Avenue, Singapore
Email: jonathan008@e.ntu.edu.sg, {zqshen, ascymiao}@ntu.edu.sg

***Abstract*—Virtual assistants are used in a wide variety of environments by different types of users. Giving users the ability to build and customize virtual assistants' skills and capabilities would enable them to create virtual assistants that can fit the needs of different scenarios. We propose a model for virtual assistants, based on Goal Net, with the aim of empowering users without programming experience to personalize and customize their virtual assistants. Goal Net separates the design of agent mental models from their low-level implementation. Developers contribute to a library of functions which can be used designers to develop functionality for their virtual assistants. The Multi-Agent Development Environment (MADE) is a graphical tool for creating Goal Net agents and allows users to easily deploy their agents for usage without the need to compile code. A case study is performed to show how Goal Net can be used to develop virtual assistant skills. The proposed model provides a foundation for future work, which would involve human computer interaction and natural language processing.**

***Index Terms*—Virtual assistant, Goal net, Agent-oriented software engineering, End-user development, Dialogue system**

I. Introduction

Modern virtual assistants are becoming increasingly intelligent and complex. Virtual assistants are used for a wide variety of purposes from handling basic tasks such as alarm setting and controlling smart home devices, to more complex applications such as assisting the elderly [1]. Given this range of use cases, enabling end-user development of assistant behaviour and capabilities would allow users to customize their assistants according to their needs. End-user development involves enabling users who are not familiar with programming to modify or create system functionality [2].

In this work, we propose a model for virtual assistants based on Goal Net, a goal-oriented methodology for agent development [3]. The Goal Nets in this work were created using the Multi-Agent Development Environment (MADE) [4], [5], which is a development environment for creating Goal Net agents. We aim to enable users less familiar with software development or agent-oriented software engineering to participate in the creation of agents by allowing them to graphically define agent behaviour using MADE.

From this point onward in this paper, we may refer to assistants' functions and capabilities as "skills". An "end-user" may refer anyone who wants to personalize their virtual assistant, including those who may not have programming experience. The rest of this paper is organized as follows. First, some related work is discussed. Then, the basics of Goal Net are introduced, and the proposed model is described. A case study is examined in which a Goal Net for a virtual assistant is created. Finally, the paper concludes with a discussion and possible directions for future work.

II. Related Work

Interest in end-user development for virtual assistant skills has been shown by both Amazon and Google, who have created platforms for developers to create voice-assisted applications for their respective virtual assistants: Amazon Alexa and Google Assistant. Amazon provides the Alexa Skills Kit[1], and Google provides Actions on Google[2]. Both development frameworks provide methods for users not familiar with programming to create their own voice-assisted applications by using templates and spreadsheets. However, they are limited to specific types of applications such as quizzes and flashcards.

Frameworks have been created to facilitate the development of spoken dialogue systems. For example, the RavenClaw dialogue management framework allows developers to create their own dialogue managers by creating dialogue task specifications [6]. These task specifications are hierarchical tree-structured representations of agents that are invoked to handle various dialogue tasks. The framework is designed to be task-independent, flexible, transparent, modular, scalable, and open-source. However, the task specifications are written in C++ and must be compiled before usage, making it difficult for end-users to personalize their virtual assistants.

Recently, there has been an effort to create open-source libraries for developing dialogue systems. For example, Rasa is a Python library aimed at providing machine learning methods for developers less familiar with dialogue systems [7]. However, such libraries are targeted at developers and are not suitable for users wihout programming experience.

Unlike previous work, we focus more on the design of agent behaviour and personalization of virtual assistants. We aim to make the design of assistant modules accessible to end-users by taking advantage of the graphical attribute of Goal Net and MADE. This helps separate the design and implementation of agents, enabling users with limited programming experience

[1]https://developer.amazon.com/alexa-skills-kit
[2]https://developers.google.com/actions

978-1-7281-4026-1/19/$31.00 ©2019 IEEE

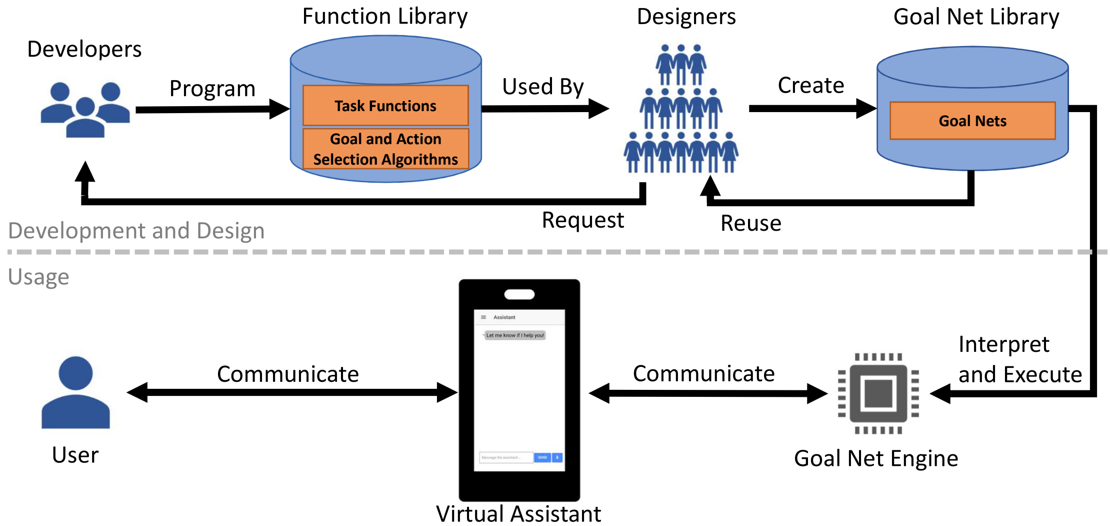

Fig. 1. Workflow using Goal Net for virtual assistants.

to create their own virtual assistant skills. MADE also allows users to deploy their agents without needing to recompile code, simplifying the process for end-users.

III. Proposed Workflow Model

A. Goal Net

In this work, Goal Net is used as the basis for developing virtual assistants. A Goal Net is a mental model of an agent, representing its goals and how it achieves those goals in dynamic environments. The basic components of a Goal Net are goals and actions. Goals can be either atomic or composite. Composite goals can be decomposed into atomic and composite goals, while atomic goals are unable to be decomposed any further. This leads to a hierarchical structure of goals, where an agent may need to accomplish sub-goals in order to achieve its overall goal. Actions represent the transition relationships between goals, describing any low-level tasks that need to be performed when transitioning from one goal to another. Goal Net allows for goal and action selection mechanisms to be defined so that an agent can decide what goals to pursue and what tasks to execute based on situational criteria [8].

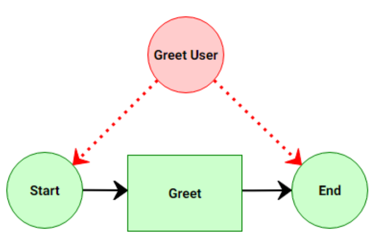

Fig. 2. A simple Goal Net consisting of a root composite goal (Greet User), two atomic goals (Start, End), and an action (Greet).

Graphically, a composite goal is represented as a red circle, an atomic goal is represented as a green circle, and an action is represented by a green rectangle, as shown in Fig. 2. The red dotted arcs indicate the start and end goals of the composite goal, and the black arcs indicate relations between goals and actions. Examples of tasks that could be associated with the Greet action may include saying “good morning” or “good afternoon” based on the time of day.

B. Workflow using Goal Net for Virtual Assistants

Our proposed workflow model is shown in Fig. 1. Developers create functions, such as goal and action selection algorithms, which are stored in a function library, which may be stored on a cloud server. Then, this library of functions is used by Goal Net designers to create Goal Nets. The designers may refer to anyone who wants to create functionality for their virtual assistant, including those who may not have programming experience. Designers may request new functions from developers and may reuse existing Goal Nets to create their own. As mentioned previously, designers can use MADE to create Goal Nets, as it contains a Goal Net Designer component. We updated the Goal Net Designer by creating mobile and web versions of the application, shown in Fig. 3, in order to accommodate a wider variety of users. Over time the function and Goal Net libraries will grow, as they are effectively crowdsourced, enabling designers to reuse many Goal Net components.

To create Goal Nets, users may follow the goal-oriented modelling methodology in [3]. Users without programming experience can focus on performing goal-oriented requirement analysis by identifying goals that they want their module to achieve, decomposing those goals into sub-goals, and identifying the actions and tasks required to transition from one goal to the next. Designers may opt to use Goal-Environment-Task (GET) cards to describe their Goal Net, which can then be passed onto developers to complete any programming required.

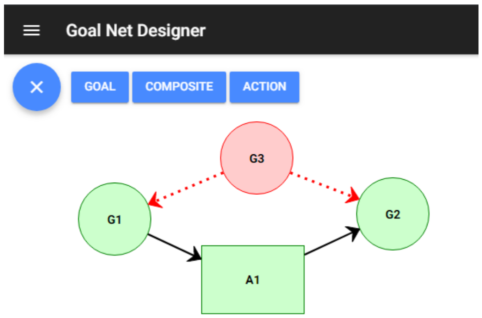

Fig. 3. A screenshot of the updated interface of the Goal Net Designer in MADE.

IV. Case Study

To demonstrate how Goal Net can be used to create virtual assistant skills, we will examine a sample Goal Net created in MADE. A mobile application was developed to enable interaction with the assistant, as shown in Fig. 4. Users can interact with the assistant by either typing a message or using spoken language. The user's utterances are displayed in blue message bubbles, and the assistant's responses in grey bubbles.

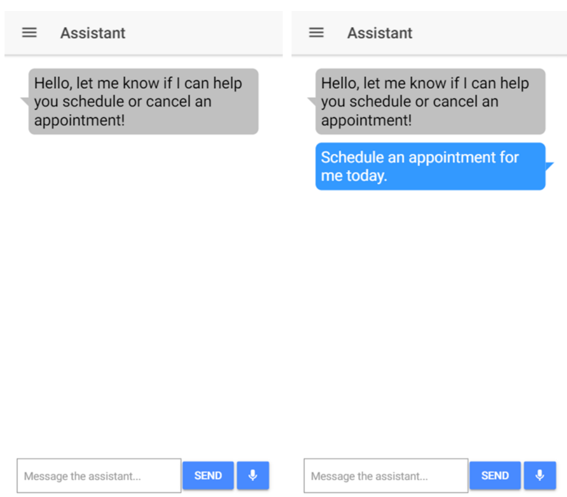

Fig. 4. Screenshots of the mobile application for the assistant.

One common function of virtual assistants is to act as a task-oriented dialogue system, where the assistant attempts to help users complete tasks such as booking flights, purchasing movie tickets, or in this example, scheduling an appointment [9]. We create an assistant that can schedule appointments by creating a Goal Net using MADE.

In this example, the assistant can either help the user create an appointment, or to cancel the user's next appointment. From Fig. 5, we can follow the overall logic of the assistant, and from Table I we can follow an example interaction between the assistant and a user. The assistant first greets the user. If the user wants to schedule an appointment, they let the assistant know by responding with a message. Once the assistant receives this response, a goal selection algorithm will be used at the "Input Received" goal to determine that the user wants to create an appointment, and so it needs to request more information from the user. In this example, the assistant requires three pieces of information to create an appointment: date, start time and end time of the appointment. The "Request Information" action contains generic functions to get the time or date from the user, which could be reused in different Goal Nets. Once the assistant has all the information it needs, it will create the appointment and tell the user that it has been created.

TABLE I
Example Usage of the Appointment Scheduler

Speaker	Utterance
Assistant:	Hello, let me know if I can help you schedule or cancel an appointment!
User:	Schedule an appointment for me today.
Assistant:	What time should I schedule your appointment?
User:	5 pm.
Assistant:	What time will your appointment end?
User:	6 pm.
Assistant:	I have create the appointment and added it to your calendar.

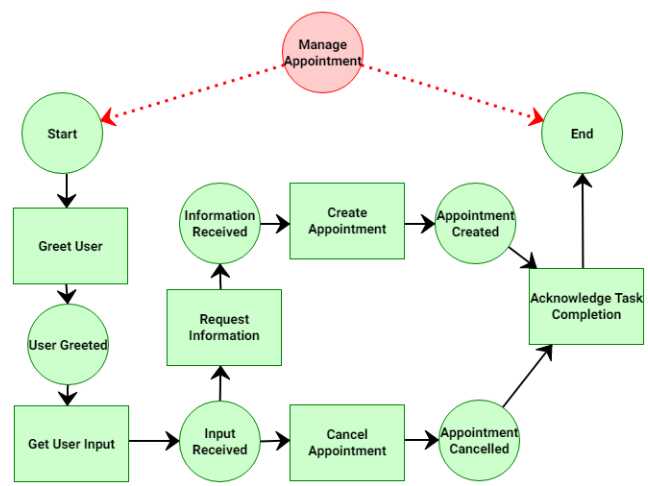

Fig. 5. The Goal Net for the Appointment Scheduler example.

V. Discussion

The modules and Goal Nets discussed in the case study demonstrate how Goal Net can be used to model various functionality and behaviour for virtual assistants. There are two main advantages that arise from our proposed model. The first is the abstraction of the low-level implementation of agents through the separation of design and development. Allowing end-users to circumvent programming by graphically creating Goal Nets using MADE opens the possibility for users to customize their own virtual assistants with less reliance on developers, while providing the expressiveness to create a wide variety of assistant skills.

Developers are required to create task functions, and goal and action selection algorithms, which would be added to the library available for designers to use in their Goal Nets. However, as the library increases, designers will have more flexibility in the types of modules they want to create.

This leads to the second advantage of our model, which is the personalization and customization of virtual assistants. With this library, designers can independently create, modify, and customize their own virtual assistants, and then easily deploy them using MADE. Goal Nets themselves are also stored in a repository, making them reusable and editable by different users. Overall, these factors can lead to expressiveness and flexibility for users to personalize their virtual assistants. This can also help shift the workload of developers onto designers, freeing up developers to spend time on other tasks.

There are a couple of other factors to note about our model. The hierarchical structure of Goal Net also allows for more complex behaviour to be created. In our example, our assistant is only able to handle appointment scheduling. The "Manage Appointment" Goal Net from our example could be included in a larger assistant Goal Net which contains multiple composite goals, one for each skill of the assistant. This could transform the assistant into a multi-domain dialogue system, which would then require a domain selector such as in [10].

Creating flexible goal and action selection algorithms and task functions for designers is key to creating a scalable system, otherwise developers will need to program many handcrafted, domain-specific functions. In our case study, we had functions to get information about times and dates from users, which could be reused in many different contexts. Such general functions make it easier for designers to select the functions they require for their tasks and can result in less development requirements from programmers.

VI. Conclusion and Future Work

Ultimately, our work aims to empower end-users to personalize their virtual assistants to fit their needs, while keeping developer overhead to a minimum. We used Goal Net and MADE as the basis of our framework, which may serve as a foundation for future work. The case study discussed in this work demonstrates how Goal Net and MADE can be used to develop virtual assistant skills. Building virtual assistants requires a combination of knowledge from both Human Computer Interaction (HCI) and Natural Language Processing (NLP), therefore future research directions could involve either of these fields.

HCI is a key factor in creating a system for end-user development. In order to ease the development process of the virtual assistant skills, MADE could be analysed using concepts from HCI, such as usability measures. Ideally, the system would be both usable by both novice and expert users while maintaining the flexibility and expressiveness to create intelligent virtual assistants. The striking a good balance between ease of use and expressiveness will require usability testing.

Another research direction involves developing general functions that can be used by designers in many different types of virtual assistant skills. For example, Bapna et al. propose a method for performing slot-filling in multiple domains, and potentially in new domains where there may be little to no training data by using natural language descriptions of slots [11].

VII. Acknowledgments

This research is supported, in part, by the Joint NTU-WeBank Research Centre on Fintech, Nanyang Technological University, Singapore.

References

[1] D. Wang, B. Subagdja, Y. Kang, and A.-H. Tan, "Silver Assistants for Aging-in-Place," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 1, IEEE, Dec. 2015, pp. 241–242.

[2] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-user development: An emerging paradigm," in *End User Development*, H. Lieberman, F. Paternò, and V. Wulf, Eds. Dordrecht: Springer Netherlands, 2006, pp. 1–8.

[3] Z. Shen, D. Li, C. Miao, R. Gay, and Y. Miao, "Goal-oriented Methodology for Agent System Development," in *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, vol. 2005, IEEE, 2005, pp. 95–101.

[4] Z. Shen, H. Yu, C. Miao, S. Li, and Y. Chen, "Multi-Agent system development MADE easy," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, pp. 4391–4392.

[5] H. Yu, Z. Shen, and C. Miao, "A goal-oriented development tool to automate the incorporation of intelligent agents into interactive digital media applications," *Comput. Entertain.*, vol. 6, no. 2, 24:1–24:15, Jul. 2008.

[6] D. Bohus and A. I. Rudnicky, "The RavenClaw dialog management framework: Architecture and systems," *Computer Speech & Language*, vol. 23, no. 3, pp. 332–361, Jul. 2009.

[7] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management," pp. 1–9, Dec. 2017. arXiv: 1712.05181.

[8] Z. Shen, C. Miao, Y. Miao, X. Tao, and R. Gay, "A Goal-oriented approach to Goal selection and action selection," *IEEE International Conference on Fuzzy Systems*, pp. 114–121, 2006.

[9] H. Chen, X. Liu, D. Yin, and J. Tang, "A Survey on Dialogue Systems: Recent Advances and New Frontiers," Nov. 2017. arXiv: 1711.01731.

[10] D. Hakkani-Tür, G. Tur, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM," in *Interspeech*, Sep. 2016, pp. 715–719.

[11] A. Bapna, G. Tur, D. Hakkani-Tur, and L. Heck, "Towards Zero-Shot Frame Semantic Parsing for Domain Scaling," Jul. 2017. arXiv: 1707.02363.