Goal-Oriented Modelling for Virtual Assistants

Jonathan Leung, Zhiqi Shen, Chunyan Miao
Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly
School of Computer Science and Engineering
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 modules 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 examine the model and show how Goal Net can be used to develop virtual assistant modules. 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], 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 interchangeably refer to assistants' "skills" as "modules". 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 different functions for a virtual assistant are developed. 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 [5]. 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 [6]. 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 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.

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

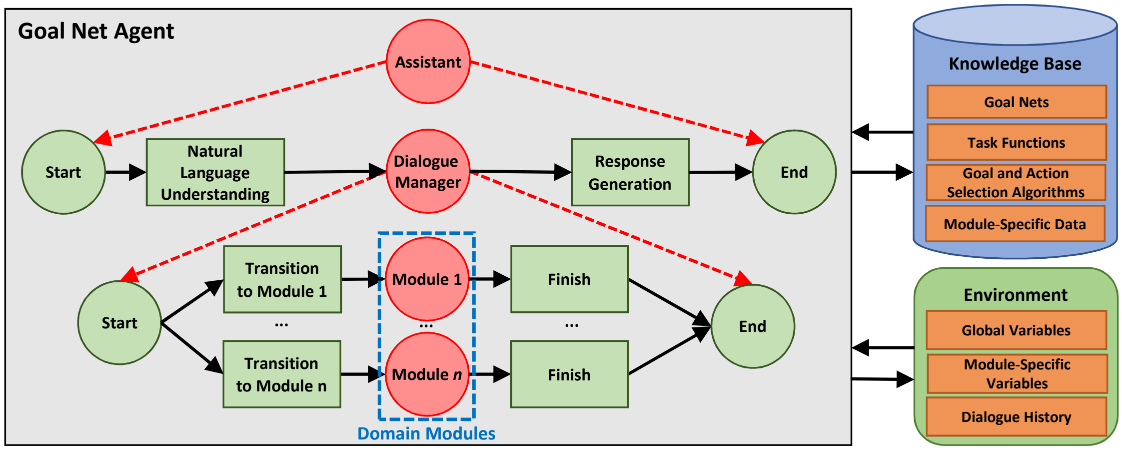

Fig. 1. Goal Net model and architecture for the proposed virtual assistant framework.

III. Proposed Model

In this work, Goal Net is used as the basis for developing virtual assistants. 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 [7].

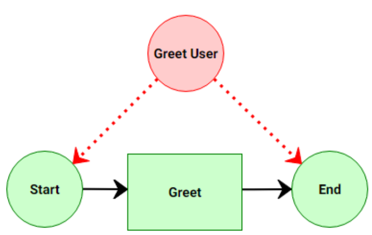

Fig. 2. A simple Goal Net consisting of a root composite goal (Greet User), two atomic goals (Start, End), and an action (Greet).

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.

The proposed model is shown in Fig. 1, where one traversal through the Goal Net represents one dialogue turn for the assistant. The model follows the general structure of task-oriented dialogue systems [8]. The methods for natural language understanding and response generation are not restricted to any particular algorithms, and can be chosen based on the desired implementation of the assistant. The dialogue manager contains a set of distributed modules $M = \{m_1, \ldots, m_n\}$ where each module is represented as a composite goal that can be achieved by the assistant and can be modelled as its own Goal Net. When the assistant receives a user utterance, the assitant first needs to determine which module needs to be selected. The structure of the composite goals will vary depending on the complexity and functionality of the module. Modules can be added as needed depending on the desired uses of the assistant, which allows for assistants to be customizable. For example, an assistant may have one module for flight booking, one for question answering in a specific domain, and one for handling general conversations with a user. Goal Nets are also editable by users, making it possible to further personalize assistants by changing modules to fit their needs.

The proposed model allows Goal Nets for different modules to be designed independently from the overall assistant Goal Net and from other modules, and each module may manage its own data and environment variables. Tasks, goal and action selection algorithms, and Goal Nets are stored in a repository which may be viewed as a library for designers to reuse in their own Goal Nets. Over time, this library is expected to grow as more virtual assistant modules are developed, as it is effectively crowdsourced. Users without programming experience can make use of existing functions in the library when designing their own modules. After designing a new module, designers can deploy their updated assistants using MADE without having to recompile any code. Using this method, end-users are potentially able to personalize their virtual assistants without needing to rely on developers to help them do so.

IV. Case Study

To demonstrate how Goal Net can be used to create different types of modules for the virtual assistant model, we consider a hypothetical scenario where a user, Mary, has a mother that lives alone. Mary is usually quite busy, but she wants to be kept up to date about the status of her mother. She uses her virtual assistant to help her achieve this, which has two modules: a data driven storyteller and an appointment scheduler. A mobile application was developed to enable interaction with the assistant, as shown in Fig. 3. 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.

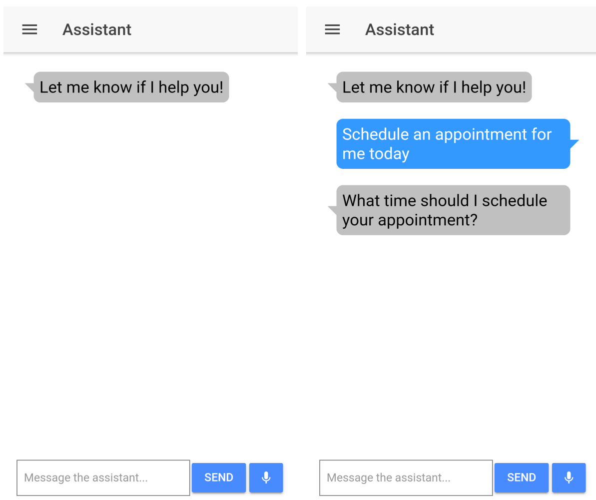

Fig. 3. Screenshots of the mobile application for the assistant.

A. Data-Driven Storyteller

Mary is busy at work and wants a quick update about how her mother is doing, so she asks her virtual assistant about the status of her mother. A sample interaction between Mary and her virtual assistant is shown in Table I, with a sample story about Mary's mother. A data-driven storytelling model based on Goal Net was proposed by Wu et al. which was developed to support social relations the between the elderly and their adult children [9]. In this case, a simple, modified version of the storyteller Goal Net is created and used as a module for the virtual assistant, as shown in Fig. 4.

TABLE I
A REQUEST FOR A STORY FROM THE ASSISTANT

Speaker	Utterance
Mary:	How is my mother doing?
Assistant:	The noise level, temperature and humidity feel about right at mother's place.
	Mother did not move around much at home.
	Mother did not host anyone at home.
	Mother cooked a lot.
	Mother did not sleep well last night.

Upon hearing Mary's initial utterance, "How is my mother doing?", the virtual assistant determines the domain and selects the storyteller module. The storyteller module retrieves the necessary data from its knowledge base to create the story. This is done in parallel, as represented by the diamond-shaped arcs. The parallel threads will synchronize at the "Process Data" action, and the assistant will interpret the data, and then generate natural language summarizations of the data.

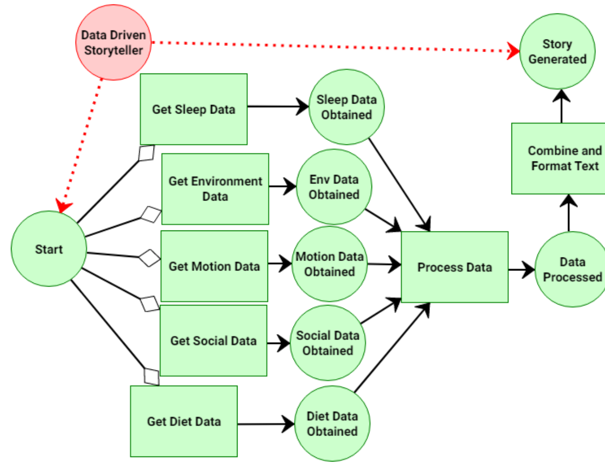

Fig. 4. Goal Net for the Data-Driven Storyteller module created using MADE.

B. Appointment Scheduler

After learning about the status of her mother, Mary decides that she wants to have lunch with her to check up her. She asks her virtual assistant to schedule a lunch appointment with her mother, shown in Table II. An appointment scheduler module was created for the assistant using MADE, shown in Fig. 5.

The assistant needs to parse the user's utterances and extract key information in a process known as slot filling [8]. The values required for the assistant to create an appointment are the date, start time, and end time of the appointment. The goal selection algorithm at the "Input Parsed" goal will pick the next action based on the current dialogue state. The user might not provide all the required information in their initial request to schedule an appointment, therefore multiple dialogue turns may be needed to successfully create the appointment.

TABLE II
EXAMPLE USAGE OF THE APPOINTMENT SCHEDULER

Speaker	Utterance
Mary:	Schedule a lunch appointment with my mother today.
Assistant:	What time should I schedule your appointment?
Mary:	12 pm.
Assistant:	What time will your appointment end?
Mary:	1 pm.
Assistant:	Ok, I'll send an invite to your mother.

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

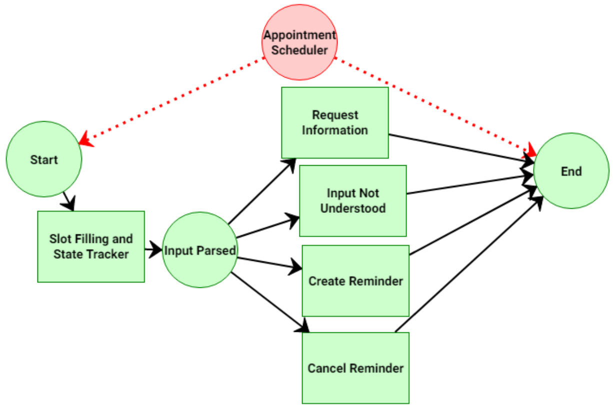

Fig. 5. The Goal Net for the Appointment Scheduler module.

to customize their own virtual assistants with less reliance on developers, while providing the expressiveness to create a wide variety of modules.

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.

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 assistants with varying skills.

Human Computer Interaction (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.

Domain selection is an important part of multi-domain dialogue systems. Recent work on multi-domain dialogue systems has proposed using recurrent neural networks to handle this task [10]. However, the assistant model proposed in this work requires the domain selector to adapt as modules are removed or added, with potentially little to no training data. Thus, an adaptive domain selector could be the subject of future work.

VII. Acknowledgments

This research is supported, in part, by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Investigatorship Programme (NRFI Award No. NRF-NRFI05-2019-0002).

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] 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.

[6] 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.

[7] 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.

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

[9] Q. Wu, Z. Shen, C. Leung, H. Zhang, Ailiya, Y. Cai, and C. Miao, "Internet of Things Based Data Driven Storytelling for Supporting Social Connections," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, IEEE, Aug. 2013, pp. 383–390.

[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.