

Cloud-based Dialog Navigation Agent System for Service Robots

Ching-Han Chen, Mu-Che Wu,* and Chien-Chun Wang

Department of Computer Science & Information Engineering, National Central University,
No. 300, Zhongda Rd., Zhongli District, Taoyuan City 32001, Taiwan (R.O.C.)

(Received February 11, 2019; accepted March 11, 2019)

Keywords: artificial intelligence, natural language processing, navigation, robotics and automation

The conventional human–robot interactions of robotic navigation systems must rely on strict language instructions and numerous button operations. In this study, a cloud-based dialog navigation agent (CDNA) system was designed for a campus navigation robot (CNR) that provides navigation services to students, school visitors, and people with impaired vision. The CDNA is based on a lightweight belief–desire–intention (BDI) software architecture (i.e., Cells, a cell-inspired efficient software framework), which is a goal-oriented and dynamic parallel framework. The proposed CDNA system has the following three primary functions: (1) conversational navigation service, (2) immediate path planning and path modification, and (3) location guide and place evaluation. The system can be applied to regional navigation guidance services such as campus tours. The Cells-based CDNA uses a natural language processing (NLP) technology to analyze the semantics of user statements and uses dialog to eliminate ambiguity in language to improve interaction with users. In this study, 15 items for three different navigation systems were evaluated, which demonstrated that the CDNA is advantageous in terms of interactivity and usability. The Cells-based CDNA can achieve an average speedup of 1.75 times in seven data sets. Therefore, the CDNA possesses the following advantages: high interactivity, high usability, and high performance.

1. Introduction

In a machine intelligence and automation technology (MIAT) lab, the campus navigation robot (CNR) plan has software and hardware designs. The software is a cloud-based dialog navigation agent (CDNA) system, whereas the hardware is a robot platform. The CDNA communicates with the robot platform through Bluetooth.

The CNR plan includes the integration of the CDNA and robot platform. The CDNA uses automatic speech recognition (ASR) and global positioning system (GPS) services with Android systems. The CDNA uses a part-of-speech tagger and question–answer services through a MIAT web service. This study focused on the design and implementation of a CDNA.

The use of intelligent service robots has increased worldwide for elderly healthcare,^(1–3) social assistance,⁽⁴⁾ poststroke rehabilitation,^(5–7) care giving,⁽⁸⁾ long-term care,⁽⁹⁾ personal

*Corresponding author: e-mail: mikewu1007@gmail.com
<https://doi.org/10.18494/SAM.2019.2326>

assistance,^(10,11) education,^(12,13) entertainment,⁽¹⁴⁾ service,^(15–17) and disaster rescue.⁽¹⁸⁾ Users can operate remote robots in campuses using the robot service network protocol.⁽¹⁹⁾

In general, robots adopt a traditional sequential software development methodology. However, the CDNA adopts a different development approach; it uses a belief–desire–intention (BDI) framework to develop the software core of the robot. A BDI software model based on human practical reasoning is a software model developed for programming intelligent agents.⁽²⁰⁾ The model provides a mechanism for distinguishing flow selection from flow execution. The Cells is a framework based on BDI and has a flow parallel mechanism.⁽²¹⁾ The autonomy of BDI is suitable for the development of perceptual software systems, such as the Internet of Things (IoT) and robot systems. Traditional BDI systems are heavyweight systems, which cannot be easily transplanted to an Android platform. Therefore, selecting between adopting nonperceived software development and using the BDI framework is a dilemma. The Cells solves this dilemma because it is a lightweight BDI framework. It can easily be transplanted into an Android platform and is capable of autonomous sensing. Furthermore, the Cells has the advantage of high performance by dynamic parallels, and thus, is used as the core engine of the CDNA.

The navigation system has been extensively studied and improved. For example, a reduced relative root mean square error of an estimated position⁽²²⁾ and improvements to the reliability and fault-tolerant capability are some of the improvements made.⁽²³⁾ In addition, several navigation systems have been proposed, including an indoor navigation system,^(24,25) a pedestrian dead reckoning system,⁽²⁶⁾ and a navigation system of an unmanned aerial vehicle.⁽²⁷⁾ However, most of the research regarding navigation systems has focused less on user experience and interaction. In this study, a new experience for users, that is, the interaction of pure dialog, was proposed. Furthermore, the ambiguous meaning of the user conversation was eliminated, and the purpose was achieved. For example, users want to visit convenience stores on campus; however, the campus has more than one convenience store. The CDNA actively interacts with the user to find a convenience store that satisfies user requirements. Figure 1 shows common



Fig. 1. (Color online) (a) Car, (b) Google, and (c) Siri navigation systems.

navigation systems, including (a) a car navigation system, (b) an OK Google navigation system, and (c) a Siri navigation system.

Dialog interaction is a comfortable approach to human–computer interactions. In recent years, considerable research has been conducted regarding dialog systems. The use of Bayesian networks improves a partially observable Markov decision process model.⁽²⁸⁾ Moreover, several scholars have studied dialog systems in different languages, including Persian,⁽²⁸⁾ Slovak,⁽²⁹⁾ and Japanese.⁽³⁰⁾ The dialog system developed using a considerable amount of training data is slightly uncoordinated in real-time applications. A dialog system with a large amount of unclassified training data has some uncoordinated dialogs in practical applications. An unrestricted dialog system is the source of these uncoordinated dialogs, and the scope involved is extremely broad. The meaning of the same sentence in different situations will not be the same, thereby generating different responses. Therefore, limiting dialog systems to situations is crucial.⁽³¹⁾ In this study, the context of conversations is restricted to the field of navigation and uses semantics and part-of-speech of vocabulary. This system can effectively respond to user requirements and provide navigation services.

Traditional navigation systems must rely on strict language instructions and numerous button operations primarily because of the uncertainty of natural language and its evident vagueness. For example, someone says to the navigation system: “I want to visit the convenience store.” In this example, the system is unaware of the preference of the user, which could be a specific chain, such as Seven Eleven, Family Mart, or Hi-Life, or simply the nearest convenience store. Natural language includes several uncertainties. Traditional navigation systems use language instructions and button operations to obtain accurate information. However, this considerably reduces the availability and usability of the system. This study focuses on the following questions:

- Q1: How do we design a navigation system that can overcome the uncertainty of natural language?
- Q2: How do we design a navigation system with high interactivity, high usability, and high performance?

In this study, the software based on CellS was developed and semantic analysis and dialog to eliminate this problem were used. The CellS-based CDNA uses natural language processing (NLP) technology to analyze the semantics of user statements. It also uses dialog to eliminate ambiguity in the language to improve interaction with users. Moreover, in this study, 15 items for three navigation systems were evaluated, indicating that the CDNA is advantageous in terms of interactivity and usability. The CellS-based CDNA can achieve an average speedup of 1.75 times in seven different datasets. The aforementioned two concerns are explained and verified in Sects. 5 and 6.

The remainder of this paper is organized as follows. Section 2 describes the dialog system, navigation system, and robot, which are from the background of this study. Section 3 presents the architecture of the CNR, and Sect. 4 the design of the CDNA. Sections 5 and 6 respectively provide comparisons and experiments concerning the CDNA. Finally, Sect. 7 presents conclusions and directions for future work.

2. Background

The design of the CDNA was based on cloud computing, NLP, BDI, and the Android platform. The design strategy used was not adopted to recreate or reinvent the existing concepts. The following concepts were adopted: Stanford Parser,⁽³²⁾ the Android platform, including ASR, GPS, and Bluetooth, the CellS software framework, and the Jersey RESTful web service.^(33,34)

Because it was first proposed by IBM in 2007,⁽³⁵⁾ cloud computing has been widely promoted by renowned companies, such as IBM, Google, Amazon, and Microsoft. In recent years, cloud computing and its services have been widely recognized and applied in several fields with a significant commercial value. For example, the cloud computing revenue of Amazon reached \$12.2 billion in 2016.⁽³⁶⁾ Furthermore, Microsoft predicts that the revenue of its cloud computing business will reach \$20 billion in 2018.⁽³⁷⁾ Tasks of a robot generally require numerous calculations, which easily exceed the computational capabilities of the robot, particularly at the core of platforms such as Android. In several studies,^(38,39) a “cloud + robot” design has been proposed to solve this problem. The CDNA adopted the same idea and added the cloud service to the design to reduce the burden on the Android platform.

The proposed web service was implemented using the Jersey RESTful framework. It differs from the design of the web development framework, which attempts to hide the underlying distributed application platform. The RESTful web service explicitly advocates loose coupling between services and their applications. In the RESTful web service, no type or static contract sharing is observed. By contrast, its application programming interface (API) is developed using the principles of content-type negotiation, hypermedia, and application protocols. The RESTful web service support seamlessly incorporates recognized and customized hypermedia-type formats into a development process. It facilitates content-type negotiation, and thus, users and services can dynamically connect to the optimal resource representation of their interactions.

NLP technology has been used in various areas.^(40,41) With NLP technology, the distance between computers and humans has been reduced, and the methods of communication between computers and humans has significantly changed. The CDNA adopted Stanford Parser for sentence segmentation and part-of-speech tagging (POST), and ConceptNet data,^(40,41) and E-HowNet to extract the notion of words.

The Chinese POST and segmentation service provided by the MIAT web service uses Stanford Parser. Stanford Parser is a new dependence analyzer that uses neural networks. It is superior to other greedy parsers in terms of accuracy and speed. The notion provided by the MIAT web service refers and integrates data from ConceptNet and E-HowNet. ConceptNet is the most commonly used common-sense database, which is currently available for free. It has a knowledge browser and an integrated NLP engine that supports numerous real-time text inference tasks, including topic generation, topic hints, semantic disambiguation and classification, impact perception, analog production, and other situation-oriented reasoning. E-HowNet proposes a general concept representation mechanism to solve the problem between string processing and conceptual processing. E-HowNet has an easy semantic structure and

decomposition properties, and thus, it can make up the required information for unknown words, phrases, and sentences in text processing.

The Cells is a lightweight BDI framework, and thus, it can easily be ported to Android. This framework has high performance because of its dynamic parallel and pipeline mechanism. Android Inc. was founded in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White in Palo Alto, California, and bought by Google in July 2005. The Android platform is a Linux-based operating system, which is currently the most popular IoT device. The CDNA uses the capabilities of the Android platform (ASR, TTS, and GPS) to construct a dialog navigation system.

3. CNR

Figure 2 displays the architecture of the CNR. The robot primarily comprises the CDNA and robot platform. The CDNA, placed on the Android platform, uses the Cells framework to implement dialog and navigation engines. The CDNA accesses the question–answer service and NLP service on the MIAT web service through the Android platform.

Figure 3 shows an interactive diagram of the operation of the CNR. The robot is a platform combining the cloud and IoT. The user communicates with the CDNA through ASR to

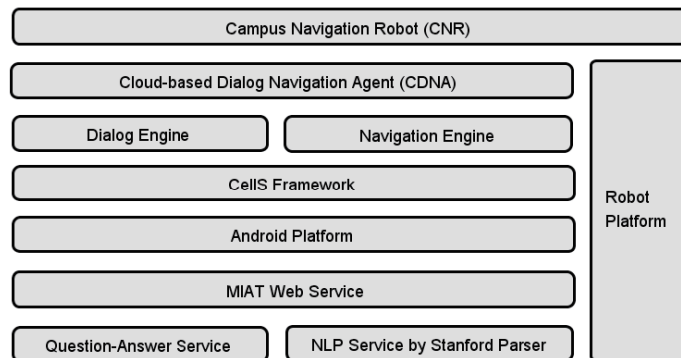


Fig. 2. Architecture of the CNR.

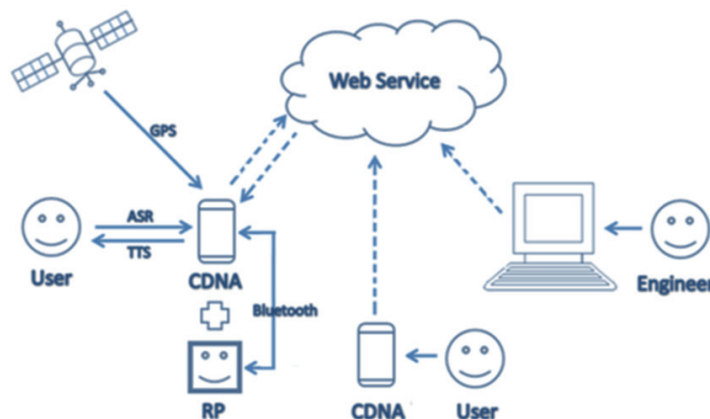


Fig. 3. Interactive diagram of the CNR operation.

validate the destination of navigation. The CDNA automatically updates information based on GPS messages and responds to users through TTS. When the CDNA generates a goal (i.e., destination), it uses Bluetooth to connect to the robot platform and controls the robot to achieve the goal. Users and engineers can construct private and public location information through the proposed web service, and the CDNA can refer to location information to navigate the robot.

3.1 Robot platform

The CNR is divided into two parts, namely, the hardware and software designs. The hardware design is the robot platform, whereas the software design is the CDNA. Figure 4 shows the real-time application of the CNR plan. The CDNA is placed above the robot platform. Because the paper focuses on the design of the CDNA, it only shows the entities of the robot platform.

3.2 CDNA

The CDNA provides two engines, namely, dialog and navigation engines. In the dialog engine, Stanford Parser and question–answer services are provided by the proposed web service. In the navigation engine, the Android platform is the infrastructure, which provides ASR, TTS, and GPS services. The notion and inference engines provide notion and inference services, respectively.

4. CDNA Design

This paper focuses on the design of the CDNA. The CDNA is divided into dialog and navigation engines. Because the CDNA is based on the Cells, the subsequent section introduces some Cells concepts and dialog and navigation engine designs.



Fig. 4. (Color online) Practical application of the CNR.

4.1 Cells design

The CDNA adopted the Cells to design the software architecture. The Cells is characterized by high performance and high scalability; it can autonomously perceive an external environment to actively change the system behavior. Each Cell is a Java class. Cells programming is based on the structure programming and object-oriented programming, and focuses on the design of the classes Cell and Plan. Figure 5 shows the dependence relationships among the Cells, object-oriented programming, and structure programming. Furthermore, Fig. 5 illustrates the relationship between the Cells and the programming. Object-oriented programming is based on a structure-oriented design. Cells programming is based on the object orientation and uses two special classes, namely, Cell and Plan. Cell determines the program flow, whereas Plan handles the flow.

Figure 6 shows the basic concept of the Cells. In general, the software can be considered a flow as shown in Fig. 6(a). Improved software with low interdependence can be divided into

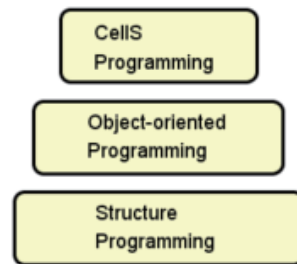


Fig. 5. (Color online) Foundation of Cells programming.

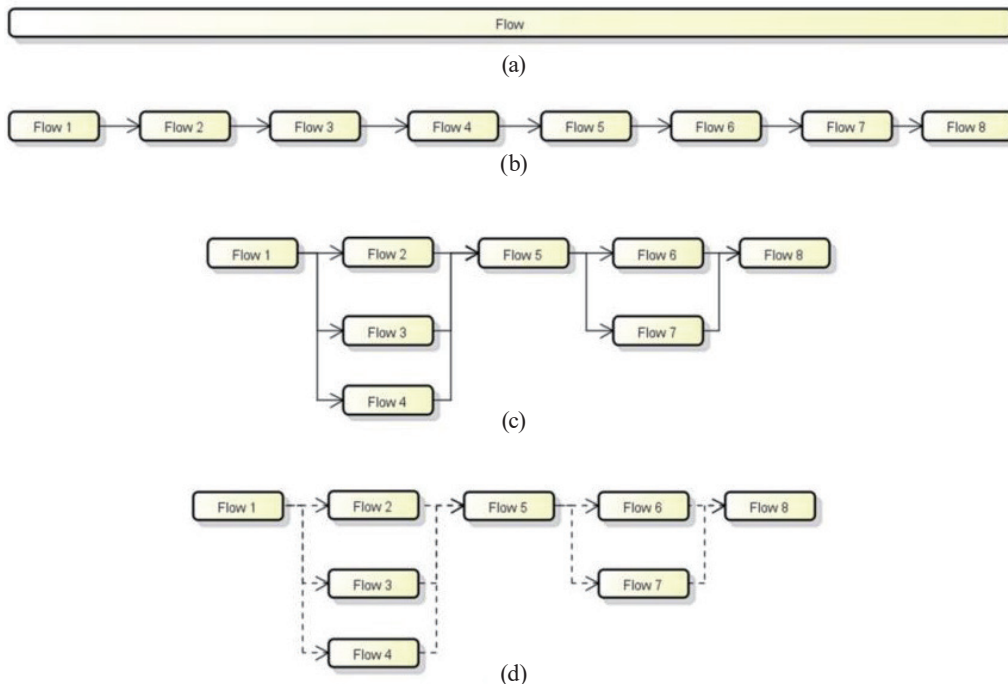


Fig. 6. (Color online) (a) Flow without modularization, (b) flow with modularization, (c) parallel flow, and (d) dynamic parallel flow.

segments of sequential flows as shown in Fig. 6(b). The flows can be parallel because they do not depend on the same data as shown in Fig. 6(c). Finally, some flows do not occur in certain situations, and we must only execute the eligible flows as shown in Fig. 6(d). The dotted line represents execution under certain circumstances; that is, a flow is not always performed. Parallel flow is a phenomenon, wherein the flow changes from Fig. 6(a) to 6(d). The process of automatically completing the parallel flow is termed the parallel flow mechanism. The implementation of Cells follows the parallel flow mechanism, which improves the software performance. Flows 2, 3, and 4 in Fig. 6(d) are not always executed. If they are executed, they must be performed simultaneously. In the Cells, the flow is termed cell. Because the cell of the Cells must be executed under certain conditions, it is suitable for systems that require interaction. The Cells is used to implement the CDNA to improve the interactivity and execution speed of the software.

In Sect. 6, the performance characteristics of software with and without the Cells are compared. In Fig. 7, all cells are connected to the CDNA by ligands, which indicate that the CDNA triggers cells through ligands. Figure 7 shows the overall message and design of the entire system. Each cell can be considered a flow. In each life cycle, different flows are dynamically composed of information in the ligands. The CDNA system dynamically generates flows (i.e., cells) to process current environmental information. Each cell also selects the optimal plan.

Figure 7 shows a visual representation of the entire system. This figure shows that Cell is categorized into three types, namely, SCell, BCell, and MCell. The responsibility of SCell is to forward the perceived message to the CDNA. For example, CSR SCell actively forwards the recognized user sentence into the CDNA through CSR ligands. The responsibility of MCell is to generate the effect on the external world. For example, Bluetooth MCell must control

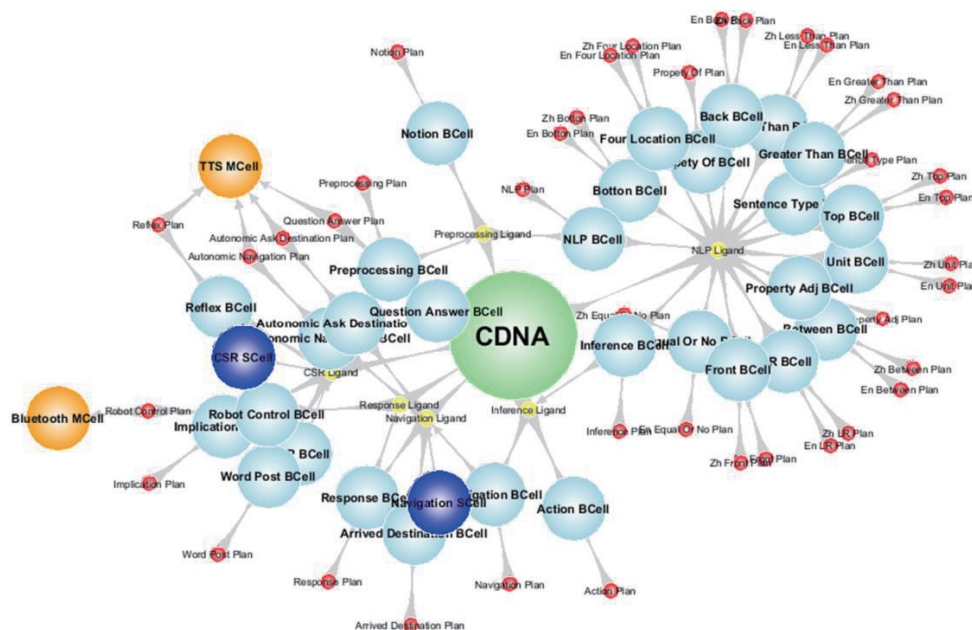


Fig. 7. (Color online) Cells design of the CDNA.

the actions of the robot platform. BCell comprises the most crucial part of the CDNA, that is, analyzing, thinking, and inferring external information to produce actions. This is consistent with the concept of the entire BDI.

Each Cell is an independent individual, depending on the ligand message to decide whether to trigger or not. When rapidly sensing a large amount of environmental information, all Cells may be triggered simultaneously to speed up the processing of external messages. The experiment described in Sect. 6 proved its acceleration effect.

4.2 Dialog engine design

The dialog engine comprises the semantic analysis of sentences and the inference of the purpose. After acquiring the user sentence through ASR, the dialog engine divides the sentence into words and determines the POS and notion for each word in Fig. 8. According to the information of the POS and notion, the dialog engine extracts the intention and goal from the entire sentence. The CDNA can now find and address two types of intentions and one goal. The two types of intentions are the response to the question and the navigation of the destination. The goal is to arrive at the destination. When the dialog model is aware of the destination, where the user wants to arrive, it transfers the goal to the navigation model using the ligand.

The dialog model can clarify fuzzy intention and achieve the goal of the user through semantic analysis and autonomous questioning. For example, the user said, “我要經過荷花池和摩斯漢堡” (I want to go by the lotus pond and Mos Burger). Figure 9 presents an analysis of this sentence.

In this study, we further focused on the design of the dialog engine, which is based on the Cells, and thus, the dialog model inherits all the Cells features. The primary purpose of the dialog engine is to analyze the semantics and determine the intention of users. It consists of five stages. From the perspective of the general software, its order is from left to right; however, in the Cells, the five stages may occur simultaneously in the same life cycle. Cells are triggered by the information of the ligand in the stage. For example, the ligand of the preprocessing stage is CSR Ligand, and its internal information can determine whether to trigger Preprocessing BCell, WordPostBCell, Implication BCell, or Reflex BCell. The functions of each stage in the dialog engine are described as follows:

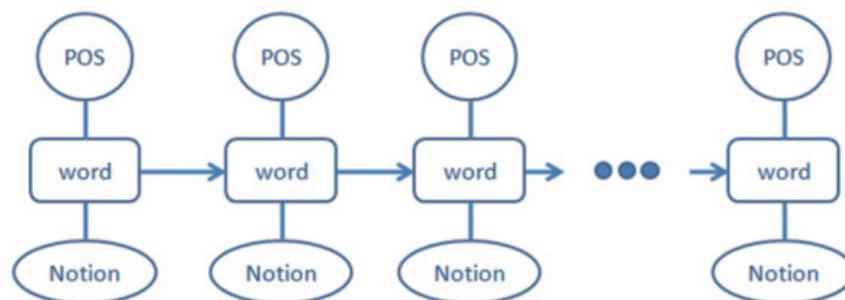


Fig. 8. Analysis format of a sentence.

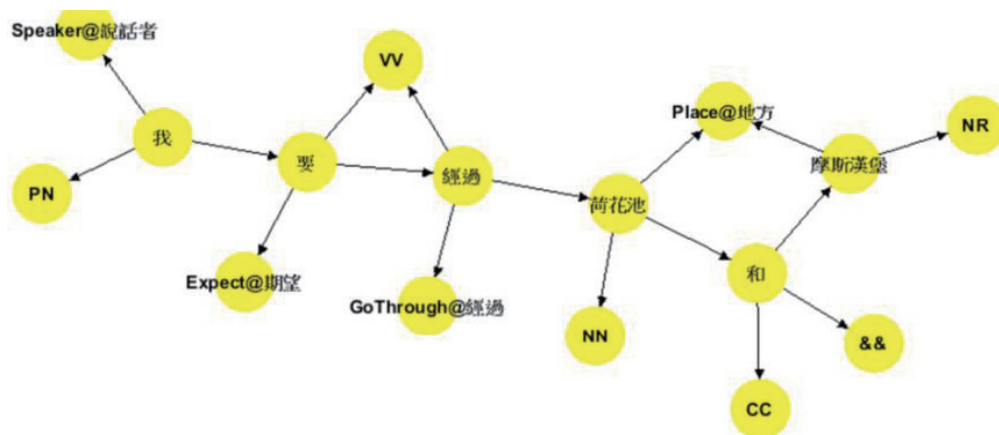


Fig. 9. (Color online) Analysis of a sentence.

- **CSR stage:** In this stage, CSR SCell actively senses whether the user has spoken to the CDNA and whether to store the recognized sentence in CSR Ligand.
- **Preprocessing stage:** This stage analyzes the sentence and generates Chinese segment words, part-of-speech, and the implication of the sentence. In this stage, if the information of the sentence is enough, the first response is obtained.
- **NLP stage:** This stage analyzes the sense of each word and stores all previous information into NLP Ligand.
- **Inference stage:** This stage uses the information of NLP Ligand to analyze the relations and information in the sentence, such as positional relations, numerical relations, and the type of sentence.
- **Action stage:** In this stage, the intention is generated from the information of Inference Ligand, and the action is generated. In the CDNA, two primary intentions are present, namely, answering questions and navigation.

The dialog engine conducts a semantic analysis of sentences through five stages and determines the user intention. In the subsequent section, the navigation engine is discussed.

4.3 Navigation engine design

To achieve the user goal, the navigation engine autonomously leads the robot to the destination. The navigation model continually and autonomously updates GPS information and checks whether the robot has reached its destination. If the robot has not reached the destination, the navigation engine transfers the relevant information through the ligand and sends a TTS response to the user. The CDNA simultaneously controls the movement of the robot using Android Bluetooth. The navigation engine ends navigation when the robot arrives at the destination or the user gives up the goal. When the user turns on the CDNA for a long period but has not stated his destination, the engine autonomously asks for the destination.

The design of the navigation engine was further explained. Moreover, the navigation engine is based on the Cells and inherits all Cells features. The navigation engine deals with different types of intentions of the CDNA during navigation. The navigation engine includes

three stages, namely, the navigation, response, and robot stages. The navigation stage actively senses whether the current CDNA is in navigation. The response stage actively interacts with the user based on different contexts in the current navigation, such as an audio guide of related attractions, interactions for reaching the destination, and the modification of a relay point during navigation. The robot stage is the route through the CDNA that actively controls the path of the robot.

In the navigation engine, the location in the sentence is used to explain the part that eliminates ambiguity. Suppose the CDNA analyzes a user who currently wishes to go to a certain destination. According to Fig. 10, four strategies are available, each of which is handled by a different Cell as follows:

- **Precise words BCell:** This Cell process obtains a precise and proprietary location. In a general navigation system, this information is required to find the destination; however, it is difficult to obtain clear information in the dialog system.
- **Similar words BCell:** This Cell process handles inaccurate location information, such as the distortion of location information in ASR.
- **Hierarchy map BCell:** This Cell handles generalized location information, such as convenience store locations. Such location information has a higher level of abstraction. In general, this information can be further divided into several categories. For example, convenience stores could be categorized as follows: Seven Eleven, OK Mart, and Family Mart. The user can select his destination from different categories that the CDNA actively proposes.
- **Similar description BCell:** This Cell handles the case wherein the location information contains additional information. This extra information can often effectively identify differences in the same category of locations and find the destination of the user.

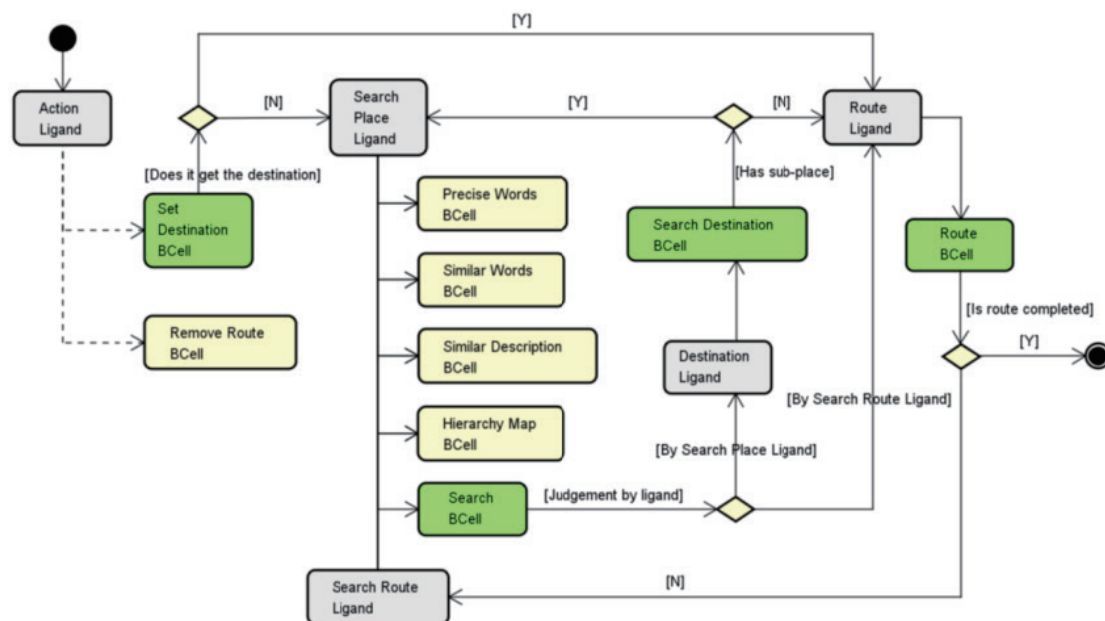


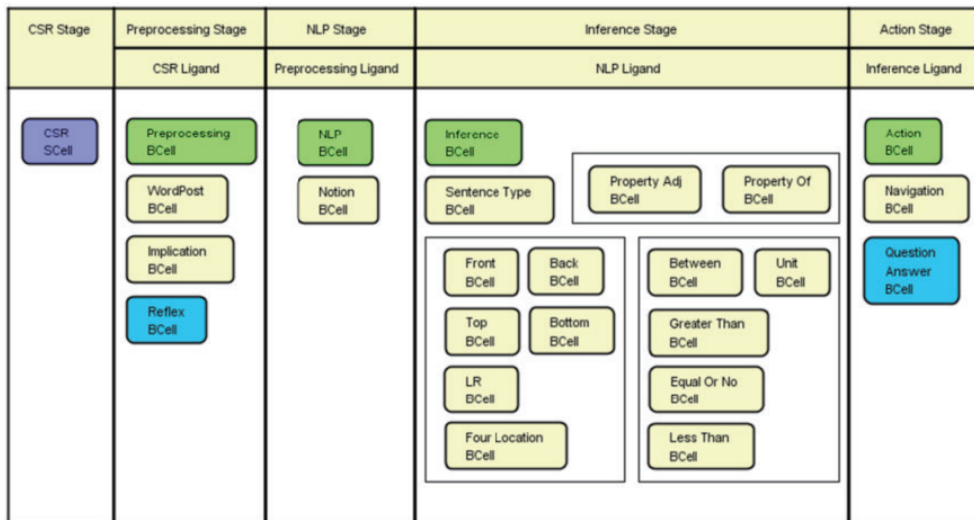
Fig. 10. (Color online) Flow for finding a position.

Through the collaboration of the four types of Cells, the CDNA can eliminate ambiguity in a user’s semantics and further analyze the destination of the user, rather than simply select the nearest location, regardless of the preference of the user. The following section verifies the ability of the CDNA in three scenarios.

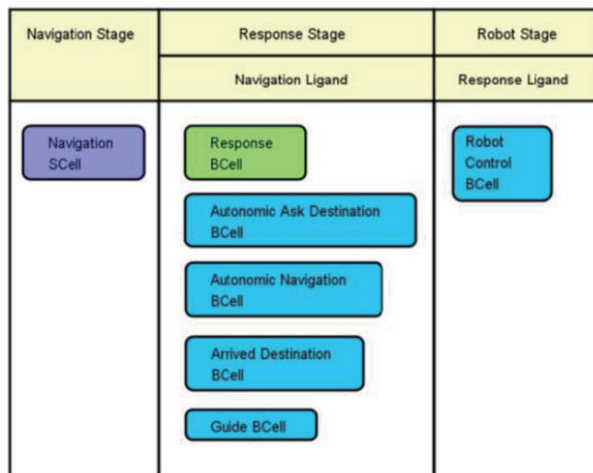
Although the dialog and navigation engines were introduced in different sections, on the basis of the Cells design, all stages of the two models may be in the same Cells life cycle. Because of Cells scalability, the dialog and navigation engines developed by different groups of people can be easily integrated without side effects.

5. System Validation and Comparison

The CDNA was developed using the Cells architecture. The CDNA is divided into two models, eight stages, six ligands, 31 cells, and 39 planes in Fig. 11. The cells are divided into



(a)



(b)

Fig. 11. (Color online) (a) Dialog flow of the CDNA and (b) navigation flow of the CDNA.

three types (i.e., SCell, BCell, and MCell, which represent the sense, brain, and motor cells, respectively). The sense cells transmit the perceived message by the ligand. When the brain cells are triggered by the ligand, they select the corresponding plan to execute as per the ligand information. Some plans trigger the motor cell to react to the external world. To evaluate the system capabilities, three types of scenarios were used. These scenarios are setting the destination, navigation, and recording a position. Figure 12 shows a screenshot of the CDNA.

5.1 System validation

In the scenario of setting a destination goal, the scenario evaluates whether the system can solve the ambiguity of natural language. The execution process is shown in Fig. 13. It shows that using dialog can solve the ambiguity of natural language and help determine the destination of the user. In the navigation scenario, the scenario evaluates whether the system can accurately navigate the user and adapt to the route change. In the real-time response scenario of the CDNA in Fig. 13, the CDNA can guide the user from the fuzzy location information (i.e., convenience store location) to determine the destination of the user (i.e., the Family Mart next to nine male dormitories).

The execution process is presented in Fig. 14, which shows that the Cells can accurately navigate the user and resolve the route changes in the general navigation system. The system cannot dynamically modify a waypoint. In the scenario where the lotus pond is removed and the Guoding Library is added, the dynamic navigation capability of the CDNA is in line with the user requirement.

In the scenario of recording a position, the scenario evaluates whether the system can autonomously provide the information regarding the environment to the user. Figure 15 shows that the Cells can autonomously provide environmental information to improve user experience. In Fig. 15, the CDNA demonstrates its capability to guide and provides users with the ability to improve the CDNA. With the guidance of the CDNA, students who are unfamiliar with the campus can become more familiar with it. Because the CDNA allows users to improve their content, other users can obtain an improved experience together.



Fig. 12. (Color online) Screenshot of the CDNA.

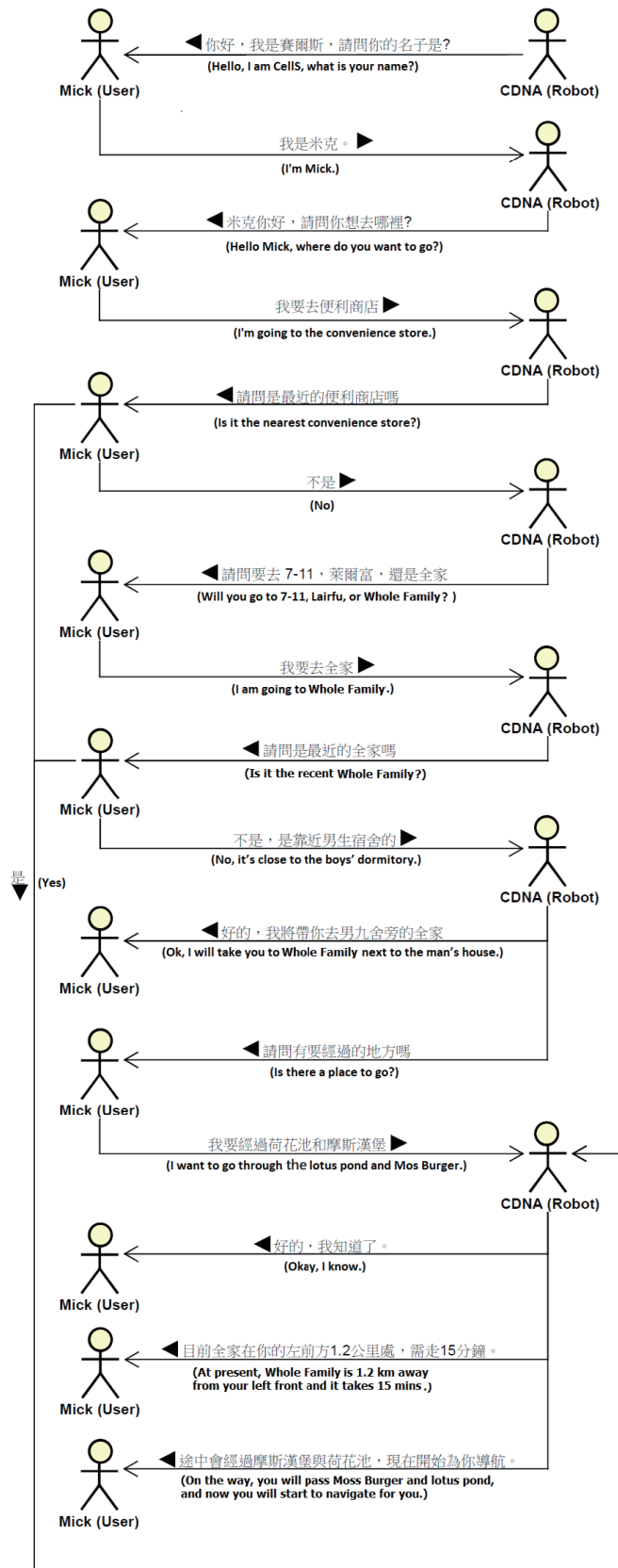


Fig. 13. (Color online) Execution process in the destination setting scenario.

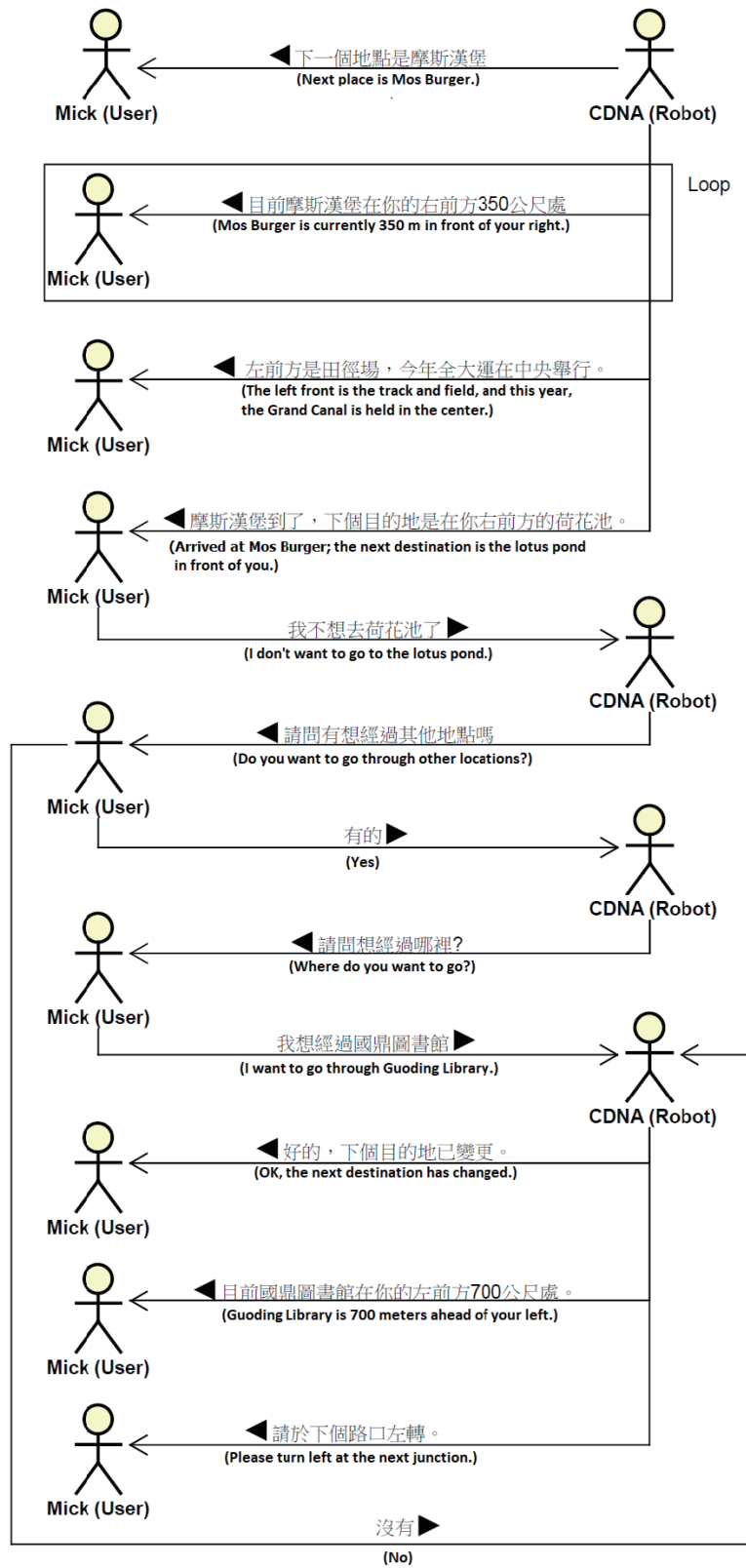


Fig. 14. (Color online) Execution process in the navigation scenario.

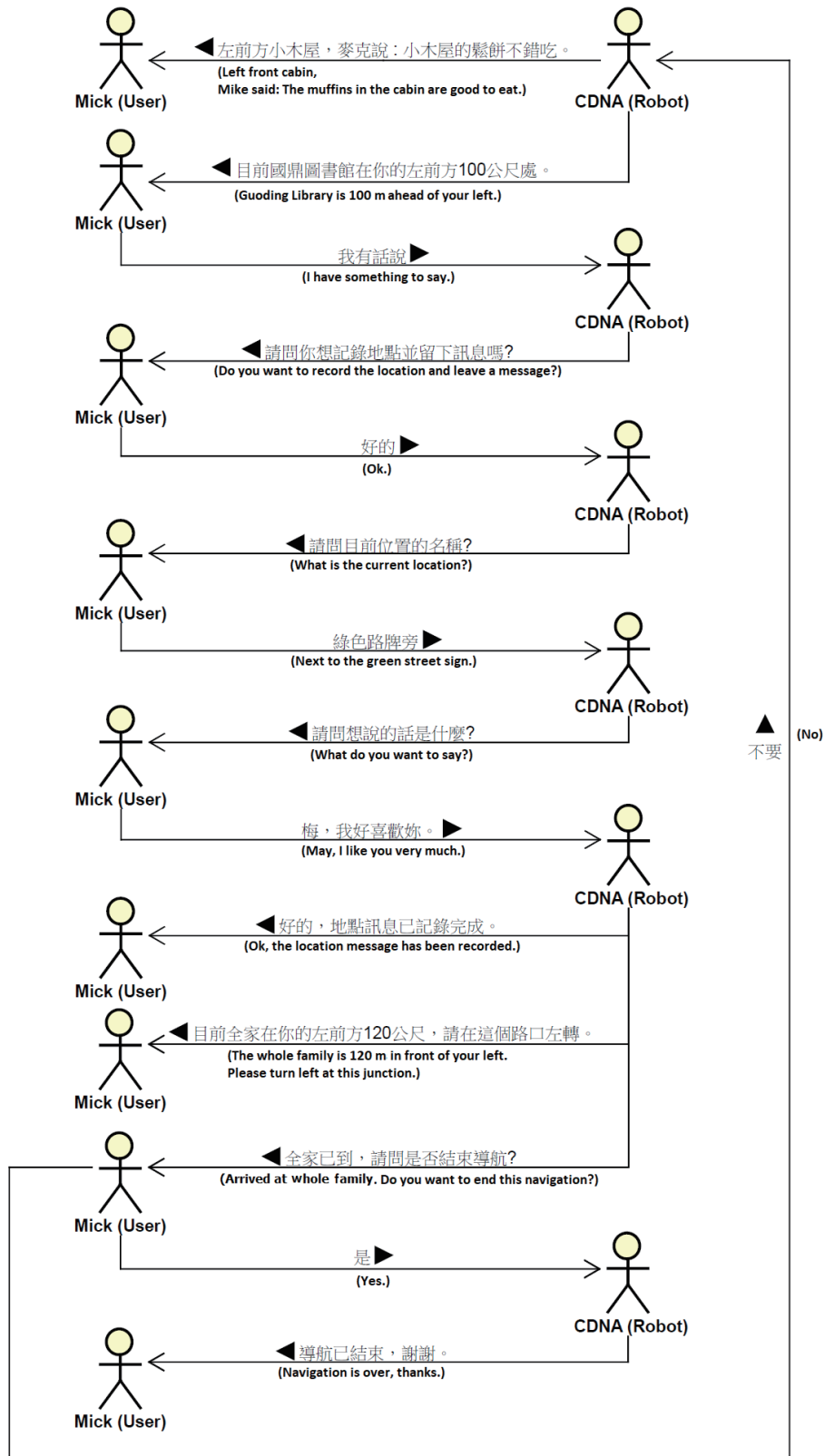


Fig. 15. (Color online) Execution process in the recording position scenario.

Figure 16 shows the travel route of the user. According to the aforementioned evaluation, the CDNA has high interactivity, high availability, and high usability. Moreover, it can solve the ambiguity of natural language. In the following section, the advantages of the CDNA are discussed by comparing the CDNA with common navigation systems.

5.2 System comparison

In this section, the CDNA is compared with the following three navigation systems: car, Google, and Siri navigation systems. Table 1 shows a comparison of the navigation systems and contains a total of 15 items. The comparison results indicate that the CDNA has an improved and friendly interface for user interaction and experience. For example, using a conversational voice input, the addition and deletion of waypoints are convenient for drivers. Solving problems related to ambiguous language considerably increases the availability of the system. Finally, the initiatives of the campus tour and environmental information not only allow new students to familiarize themselves with the campus, but can also help blind people to adapt to the campus environment.

6. Experiment

In the experiment, each task set included 100 tasks that arrived in our system simultaneously. Table 2 shows the configuration of the evaluation platform. We adopted a Qualcomm Snapdragon 810 processor for our computation platform, which has 8 cores, each of which has a clock rate of 1.5 GHz.

Figure 17 shows the execution times of the software with and without the Cells in seven datasets. The figure also shows that the software with the Cells has a shorter execution time and thus superior performance.

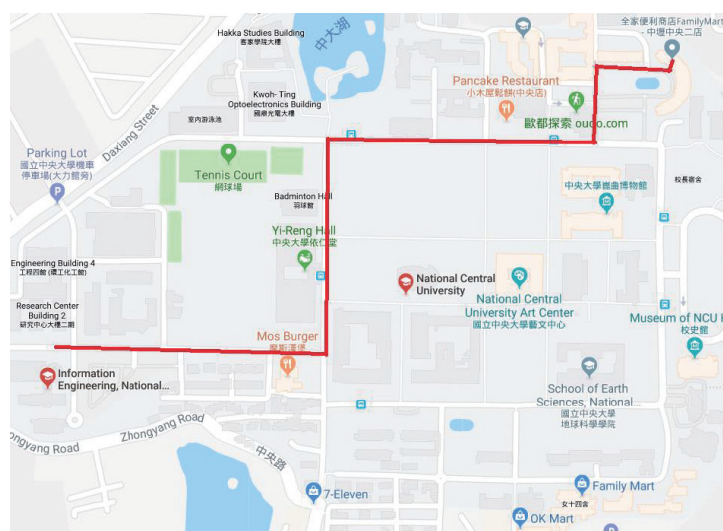


Fig. 16. (Color online) Real user route in the National Central University.

Table 1
Comparison of the navigation systems.

Navigation system	Car	Google	Siri	CDNA
1. Main input	Button	Voice	Voice	Voice
2. Interactive mode	Command	Command	Dialog	Dialog
3. Does it require a button input?	Yes	No	No	No
4. Does it have a voice input?	Yes	Yes	Yes	Yes
5. Do you require manual operation when using it?	Yes	No	No	No
6. Do you need a trigger word every time you enter your voice?	No	Yes	Yes	No
7. Does it require entering a detailed address?	Yes	No	No	No
8. Can users add location information?	Yes	Yes	No	Yes
9. Can it solve the ambiguity of sentences? (e.g., convenience store)	No	No	No	Yes
10. Can it add a waypoint in navigation?	No	No	No	Yes
11. Can it remove a waypoint in navigation?	No	No	No	Yes
12. Will it actively inform users about environmental information?	No	No	No	Yes
13. Does it have an active guidance function?	No	No	No	Yes
14. Can it be used by blind people?	No	No	No	Yes
15. Is it easy to augment new features? (e.g., connect to a robot)	No	No	No	Yes

Table 2
Hardware configurations.

Configuration	Qualcomm Snapdragon 810
Number of cores	8
Clocks	1.5 GHz
Memory size	3 GB
OS	Android 6.0 Marshmallow

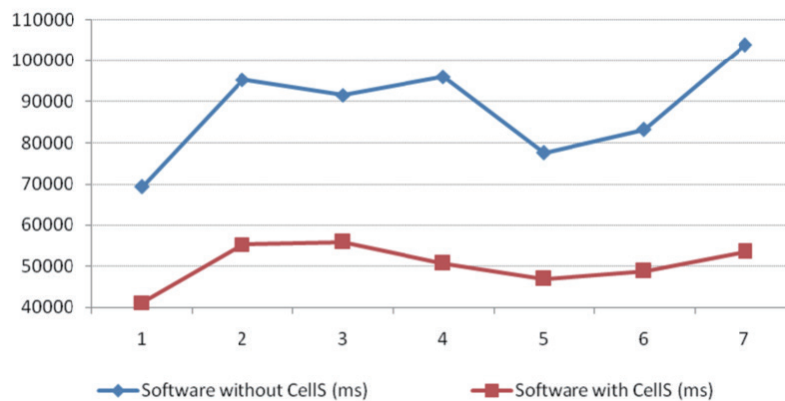


Fig. 17. (Color online) Execution times of software with and without CellS in different task sets.

The speedup in terms of execution time was evaluated using Qualcomm Snapdragon 810. Table 3 shows the evaluation results of the execution time speedups. The execution time of software with the CellS architecture was less than that of the same software without the architecture. The CellS had improved performance because the results indicated that its average speedup was 1.75 times and its maximum speedup was 1.94 times that of the same software. The formula of speedup is as follows:

Table 3
Speedup (time) of Qualcomm Snapdragon 810.

Task set	Software without CellS (ms)	Software with CellS (ms)	Speedup
1	69473.00	41146.00	1.69
2	95372.00	55205.00	1.73
3	91583.00	55833.00	1.64
4	96072.00	50634.00	1.90
5	77652.00	46955.00	1.65
6	83296.00	48736.00	1.71
7	103904.00	53446.00	1.94

$$\text{speedup} = (\text{the time without the CDNA}) / (\text{the time with the CDNA}), \quad (1)$$

$$\text{average speedup} = (\text{total speedup}) / (\text{total number of task sets}). \quad (2)$$

7. Conclusions

The CellS-based CDNA uses NLP technology to analyze the semantics of user statements and uses dialog to eliminate ambiguity in the language to improve interaction with users. In this study, 15 items for three different navigation systems were evaluated, demonstrating that the CDNA is advantageous in terms of interactivity and usability. The CellS-based CDNA can achieve an average speedup of 1.75 times in seven different datasets. Traditional navigation systems must rely on strict language instructions and button manipulation. The CDNA applied multiple rounds of dialog to resolve ambiguities in natural language. The system verification and experiment indicated that the CDNA has high interactivity, high usability, and high performance.

The CellS has high scalability, which is crucial for software engineering. The system flow changes when cells are designed at the beginning. Refactoring a completed system flow in traditional software development is an extremely difficult task. However, in the CellS, the refactoring of the system flow just need to modify the related Cell classes. This advantage can reduce the side effects of the software development process, thereby reducing the cost of software development. Our future studies will use the dialog engine and CellS for different AI system applications.

Acknowledgments

The authors are grateful for the financial support of the Ministry of Science and Technology of the Republic of China (MOST 106-3114-E-008-006, 106-2221-E-008-004, 107-2221-E-008-035, and 107-2218-E-002-056).

References

- 1 S. Koceski and N. Koceska: J. Med. Syst. **40** (2016) 121. <https://doi.org/10.1007/s10916-016-0481-x>
- 2 M. Mast, M. Burmester, B. Graf, F. Weisshardt, G. Arbeiter, M. Španěl, Z. Materna, P. Smrž, and G. Kronreif: in Ambient Assisted Living, Ed. R. Wichert and H. Klausning (Springer, Cham, Switzerland, 2015) p. 15. https://doi.org/10.1007/978-3-319-11866-6_2

- 3 E. Sato-Shimokawara, S. Nomura, Y. Shinoda, H. Lee, T. Takatani, K. Wada, and T. Yamaguchi: 24th IEEE Int. Symp. Robot and Human Interactive Communication (RO-MAN, 2015). <https://doi.org/10.1109/ROMAN.2015.7333648>
- 4 C. Jayawardena, I. Kuo, E. Broadbent, and B. A. MacDonald: IEEE Syst. J. (2016) 1056. <https://doi.org/10.1109/JSYST.2014.2337882>
- 5 H. M. Gross, A. Scheidig, K. Debes, E. Einhorn, M. Eisenbach, S. Mueller, T. Schmiedel, T. Q. Trinh, C. Weinrich, T. Wengefeld, A. Bley, and C. Martin: Auton. Robot. **41** (2017) 679. <https://doi.org/10.1007/s10514-016-9552-6>
- 6 A. Tapus, C. Țăpuș, and M. J. Mataric: Intell. Service Robot. **1** (2008) 169. <https://doi.org/10.1007/s11370-008-0017-4>
- 7 K. Swift-Spong, E. Short, E. Wade, and M. J. Mataric: IEEE Int. Conf. Rehabil. Robot. (ICORR) (IEEE, 2015) 764. <https://doi.org/10.1109/ICORR.2015.7281294>
- 8 R. Liu and X. Zhang: Int. J. Adv. Robot. Syst. **13** (2016) 1. <https://doi.org/10.1177/1729881416662780>
- 9 K. Gerling, D. Hebesberger, C. Dondrup, T. Kortner, and M. Hanheide: Zeitschrift für Gerontologie und Geriatrie **49** (2016) 288 (in German). <https://doi.org/10.7748/eldc.9.6.3.s1>
- 10 Y. Zhang, Y. Hu, P. Zhang, and W. X. Zhang: Int. J. Recent Trends Human Comput. Interaction (IJHCI) **5** (2014) 40. <https://doi.org/10.4018/978-1-59140-562-7.ch025>
- 11 E. Clotet, D. Martinez, J. Moreno, M. Tresanchez, and J. Palacin: Sensors **16** (2016) 610. <https://doi.org/10.3390/s16050610>
- 12 N. Reich-Stiebert and F. Eyssel: Int. J. Social Robot. **7** (2015) 875. <https://doi.org/10.1007/s12369-015-0308-9>
- 13 A. Causo, G. T. Vo, I. M. Chen, and S. H. Yeo: Robotics and Mechatronic (Springer, Cham, Switzerland 2016) p. 75. https://doi.org/10.1007/978-3-319-22368-1_8
- 14 I. Aaltonen, A. Arvola, P. Heikkilä, and H. Lammi: Proc. Companion 2017 ACM/IEEE Int. Conf. Human-Robot Interaction (2017) 53–54. <https://doi.org/10.1145/3029798.3038362>
- 15 M. E. Foster, R. Alami, O. Gestranus, O. Lemon, M. Niemelä, J. M. Odobez, and A. K. Pandey: ICSR 2016. LNCS (LNAI), Ed. A. Agah, J. J. Cabibihan, A. M. Howard, M. A. Salichs, and H. He (Springer Cham, 2016) Vol. 9979, p. 753. https://doi.org/10.1007/978-3-319-47437-3_74
- 16 Y. Kim and W. C. Yoon: IEEE Trans. Syst. Man Cybern. Syst. **44** (2014) 981. <https://doi.org/10.1109/TSMC.2014.2298214>
- 17 S. Schiffer, A. Ferrein, and G. Lakemeyer: Intell. Service Robot. **5** (2012) 259. <https://doi.org/10.1007/s11370-012-0118-y>
- 18 H. S. Wang, J. Ren, and X. Li: 9th Int. Conf. Human Syst. Interactions (HSI, 2016) 216–222. <https://doi.org/10.1109/HSI.2016.7529634>
- 19 T. Kusu, Y. Ito, T. Kida, T. Shimada, M. Takahashi, Y. Nomoto, Y. Tsuchiya, M. Narita, and Y. Kato: 27th Int. Conf. Advanced Information Networking and Applications Workshops (2013). <https://doi.org/10.1109/WAINA.2013.160>
- 20 A. S. Rao and M. P. Georgeff: Proc. 2nd Int. Conf. Principles of Knowledge Representation and Reasoning (1991). https://doi.org/10.1007/978-3-7091-2670-7_13
- 21 M. C. Wu and C. H. Chen: IEEE Int. Conf. Consumer Electronics-Taiwan (IEEE, 2018).
- 22 N. Davari and A. Gholami: IEEE Sens. J. **17** (2017) 1061. <https://doi.org/10.1109/JSEN.2016.2637402>
- 23 W. Jiang, Y. Li, and C. Rizos: IEEE Trans. Intell. Transp. Syst. **18** (2017) 103. <https://doi.org/10.1109/TITS.2016.2562700>
- 24 Q. Zeng, J. Wang, Q. Meng, X. Zhang, and S. Zeng: IEEE Sens. J. **18** (2018) 363. <https://doi.org/10.1109/JSEN.2017.2764509>
- 25 U. Rehman and S. Cao: IEEE Trans. Hum.-Mach. Syst. **47** (2016) 1. <https://doi.org/10.1109/THMS.2016.2620106>
- 26 H. Ju, S. Y. Park, and C. G. Park: IEEE Sens. J. **18** (2018) 6756. <https://doi.org/10.1109/JSEN.2018.2847356>
- 27 E. Fresk, G. Nikolakopoulos, and T. Gustafsson: IEEE Trans. Control Syst. Technol. **25** (2017) 192. <https://doi.org/10.1109/TCST.2016.2542022>
- 28 H. Mahmoudi and M. M. Homayounpour: Int. Symp. Artificial Intelligence and Signal Processing (AISP, 2015). <https://doi.org/10.1109/AISP.2015.7123530>
- 29 S. Ondas and J. Juhar: Int. Conf. Syst. Signals and Image Processing (IWSSIP, 2016). <https://doi.org/10.1109/IWSSIP.2016.7502774>
- 30 E. Seto and N. Kitaoka: Int. Conf. Advanced Informatics, Concepts, Theory, and Applications (ICAICTA, 2017). <https://doi.org/10.1109/ICAICTA.2017.8090987>
- 31 D. Ramachandran, P. Z. Yeh, W. Jarrold, B. Douglas, A. Ratnaparkhi, R. Provine, J. Mendel, and A. Emfield: IEEE Spoken Language Technology Workshop (SLT, 2014). <https://doi.org/10.1109/SLT.2014.7078642>

- 32 D. Chen and C. Manning: Proc. 2014 Conf. Empirical Methods Natural Language Process. (EMNLP, 2014) 740–750. <https://doi.org/10.3115/v1/D14-1082>
- 33 M. Hadley and P. Sandoz: JSR 311: JAX-RS (2009). <http://jcp.org/en/jsr/detail?id=311>
- 34 M. Hadley, S. Pericas-Geertsen, and P. Sandoz: Proc. First Int. Workshop on RESTful Des. (2010). <https://doi.org/10.1145/1798354.1798378>
- 35 IBM: IBM Introduces Ready-to-use Cloud Computing (2007). <https://www-03.ibm.com/press/us/en/pressrelease/22613>
- 36 J. Bort: Amazon's Massive Cloud Business Hit Over \$12 Billion in Revenue and \$3 Billion in Profit in 2016, 2017. <http://www.businessinsider.com/amazons-cloud-businesshits-over-12-billion-in-revenue-2017-2>
- 37 M. Rosoff: Microsoft Vows to Have \$20 Billion Incloud Revenue by 2018, 2015. <http://www.businessinsider.com/microsoft-20-billion-cloud-revenue-and-1-billion-windows-10-pcs-by-2018-2015-4/?IR=T#iz25lv5lyAbEVkd3.97>
- 38 A. Koubaa and B. Qureshi: IEEE Access **6** (2018) 13810. <https://doi.org/10.1109/ACCESS.2018.2811762>
- 39 J. J. Kuffner: Proc. IEEE-RAS Int. Conf. Humanoid Robot (2010) 176–181.
- 40 H. Liu and P. Singh: BT Technol. J. **22** (2004) 211. <https://doi.org/10.1023/B:BTTJ.0000047600.45421.6d>
- 41 W. T. Chen, S. C. Lin, S. L. Huang, Y. S. Chung, and K. J. Chen: Proc. 23rd Int. Conf. Comput. Linguistics: Demonstrations (2010).

About the Authors



Ching-Han Chen received his D.E.A degree in informatique, automatique et productique in 1992 and his Ph.D. degree in 1995 from the Franche-Comte University, France. He was an associate professor in the Department of Electrical Engineering at I-Shou University from 1995. Since 2006, he has been an associate professor in the Department of Computer Science and Information Engineering, National Central University, Taiwan. His research interests include embedded system design, multimedia signal processing, and robotics. He is currently researching machine intelligence for industrial Internet-of-Things applications. (pierre@g.ncu.edu.tw)



Mu-Che Wu received his master's degree in computer science from the National Changhua University of Education, Changhua, Taiwan in 2009. He is currently a Ph.D. student in the Department of Computer Science and Information Engineering at National Central University, Taiwan. His graduation thesis was entitled "A Hybrid Method of Text Mining," which was about data mining, and was published in 2009 (Conference of Digital Technology and Innovation Management). He is currently researching agent-based environments and agile methodology. (mikewu1007@gmail.com)



Chien-Chun Wang received his B.S. degree in computer science and information engineering and his M.S degree in electronic engineering from National Quemoy University, Taiwan, in 2012 and 2014, respectively. He is currently pursuing a Ph.D. degree in the Department of Computer Science and Information Engineering at National Central University, Taiwan. His research interests include embedded system design, wearable devices, and robotic control. (patrick@g.ncu.edu.tw)