TOWARDS AN ONTOLOGY OF COGNITIVE ASSISTANTS

Maier, Torsten; Menold, Jessica; McComb, Christopher

The Pennsylvania State University

ABSTRACT

Cognitive assistants such as IBM Watson and Siri are at the forefront of social and technological innovation and have the potential to solve many unique problems. However, the lack of standardization and classification within the field impedes critical analysis of existing cognitive assistants and may further inhibit their growth into more useful applications. This paper discusses the development of an ontology, its classes, and subclasses that may serve as a foundation for defining and differentiating CAs. Specifically, the four suggested classes include: learning, intelligence, autonomy, and communication. Various assistants are described and categorized using the proposed system. Our novel ontological framework is the first step towards a classification system for this burgeoning field.

Keywords: Artificial intelligence, Case study, Ontologies, Cognitive Assistants

Contact:
Maier, Torsten
The Pennsylvania State University
Industrial Engineering
United States of America
torstennamaier@outlook.com

Cite this article: Maier, T., Menold, J., McComb, C. (2019) 'Towards an Ontology of Cognitive Assistants', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.270

1 INTRODUCTION

Cognitive assistants (CAs), like Apple's Siri or Amazon's Alexa, are members of a revolutionary class of technologies that provide humans with a new way to interact with technology and information. However, developers are only just beginning to explore what these assistants are capable of. Currently, 20% of all US households with Wi-Fi have a smart speaker (Bernard, 2018), a common embodiment for a CA. Moving forward, CAs will likely become even more popular and pervasive due to an increase in consumer trust in the product, expansion of the industry, and a rise in disposable income (Kumar and Rasal, 2018). Moreover, the development of this technology is outpacing the research and regulation of its own field. CAs are being turned to for solutions in many fields, from health care (Ferguson *et al.*, 2010) to education (Coronado *et al.*, 2018). This technology has grown to a capacity which requires regulation and governing organizational bodies, especially as CAs are used to address issues for sensitive populations, such as the elderly or disabled. However, regulatory bodies require clarity and classification of the domain to develop adequate policies. To aid in the standardization and classification of CAs, a novel ontology is proposed in this paper.

Design is a complex activity and thus designers often rely on experience rather than defined tools with narrow uses. However, artificial intelligence (AI) may one day provide broadly applicable tools that would be useful to designers. The divide between AI and engineering design could be bridged by CAs, which can be used in shared design spaces to enable the cooperation and communication of human designers (Wu *et al.*, 2006). There have already been several attempts at using CAs in engineering design (Shen, Norrie and Barthès, 2010). For example, ARDECO captures design episodes to be adapted and reused in related computer-aided design activities (Champin, 2003).

To design efficiently, designers need goals and constraints to narrow the possible solution space. An overly broad solution space may take too much time and resources to solve to be worth the return. The proposed ontology provides designers with a road map of which level of learning, intelligence, autonomy, and communication they should expect to design for to solve certain problems. We further situate our understanding of CA research in engineering design using Concept-Knowledge (C-K) theory (Hatchuel and Weil, 2009), which frames the dynamic process of knowledge transformation in design. We posit that most research to date has generated specific instantiations of CAs, or propositions that can be modelled in the C-Space (concept space). We argue that the current work develops new information regarding the design of CAs and thus extends the K-Space (knowledge space) of the design problem.

The roots of current cognitive assistants can be traced back more than eight decades. In 1945 Vannevar Bush envisioned a device that would store all communication and printed work (Bush, 1966). This device was named the Memex; it influenced the design of future hypertext and expert systems (for example, the World Wide Web). Five years later, Alan Turing developed a test of "a machine's ability to exhibit intelligent behaviour," recognized today as the Turing Test. In 1956, Arthur Samuel developed the first self-learning program and an early example of artificial intelligence, "The Samuel Checkers-playing Program," and coined the term "machine learning" (Samuel, 1959). In the early 1960s, MIT's Artificial Intelligence Laboratories developed the first natural language processing computer program or chatbot, as we call them today, named Eliza (Weizenbaum, 1966). Eliza used rules and scripts to respond to participants with nondirectional questions. DENDRAL was one of the first expert systems, developed in the 1960s to aid organic chemists to identify unknown organic molecules (Buchanan, Sutherland and Feigenbaum, 1966). MYCIN was developed based on DENDRAL; it used backward chaining and artificial intelligence to identify bacteria causing infection and advise antibiotics (Shortliffe, 1975). In 1994, Wildfire communications developed a speech based electronic secretary that could route calls and handle messages which pioneered the fields of intelligent software assistance and voice user interfaces. In the early 2000's DARPA developed the Personalized Assistant that Learns (PAL) program. This program led to many advances in machine learning and reasoning technologies; most notably, the Cognitive Assistant that Learns and Organizes (CALO) (Berry *et al.*, 2017). CALO has the following six functions: organizing and prioritizing information, preparing information artifacts, mediating human communications, task management, scheduling and reasoning in time, and resource allocation.

Fast forward to 2011, when Apple released Siri, a mobile natural language interface for their iOS and macOS operating systems. Today, Siri is the most popular mobile virtual assistant with roughly 40 million monthly unique users (as of May 2017) (Perez, 2018). Although not the first, Siri kicked off an

https://doi.org/10.1017/dsi.2019.270 Published online by Cambridge University Press

arms race of mobile virtual assistants. Samsung (S-Voice, Bixby), Amazon (Alexa), Windows (Cortana), and Google (the Google Assistant) all responded in turn with their own versions. All these assistants provide similar features that aid in daily life. Additionally, smaller competing companies have emerged with niche products. These include Djingo (a smart speaker like the Google Home Mini), Duer (an assistant with a screen like the Amazon Echo Show), and Clova (a smart speaker company specializing in speakers for kids).

CA platforms enable both web developers and computer/mobile application developers to easily create, deploy, and use CAs (Kuznar *et al.*, 2014). This foundation is provided by companies such as James, a CA platform with modules in banking, telecom, human resources, and insurance, to name a few (Boost, 2018). CA platforms can be used for a variety of services ranging from call centers to quick-service restaurants. James processes user input (though web-based chat windows) using natural language processing. Then it predicts the user's intent through deep learning methodologies including convolutional neural networks and recurring neural networks. Finally, it dynamically responds based on pre-defined text and external application programming interfaces.

To date, a standard schema to structure information and domain knowledge relevant to CAs has not been developed in academia or industry. It is imperative that this class of technologies be studied to develop a schema that can be used to structure ongoing and future research. Current CA research growth has not matched the growth of CA products being developed in industry. This is especially concerning for products directed at sensitive populations where the impact on mental health and social well-being are not studied and understood. As organized governing bodies begin to regulate these products, they will require a clear picture of the differing categories and properties that make up CAs. To begin this process, this paper proposes the first classes and subclasses of an ontology and applies them to current CAs.

The rest of this paper is organized as follows. Section 3 discusses the ontological approach, the definition of CAs, and the identification of classes. Section 4 further defines the classes and subclasses of the ontology. Section 5 classifies current CAs using the proposed ontological framework. Section 6 identifies future work and limitations.

2 SCOPE AND CLASS IDENTIFICATION

Ontologies have multiple advantages, such as 1) a shared understanding of the structure of information 2) the creation of reusable domain knowledge 3) explicitly stated domain assumptions and 4) separate domain and operational knowledge (Chang and Rai, 2010). According to Chang *et al.* (2010), ontology development proceeds through four characteristic stages: 1) definition of scope and domain knowledge; 2) development of classes, relations, and subclasses based on similar ontologies; 3) definition of slot relations; and 4) generation and iteration using instances. Ontologies can either be developed top-down, bottom-up, or mixed. In top-down, the most general classifications are identified first, subclasses of classes and subclasses are then generated from existing entries. In bottom-up, instances are collected and grouped into similar subclasses. These subclasses are then grouped into further subclasses or classes. For the purposes of this paper, a top-down strategy is applied.

There is no single, unifying definition for "cognitive assistant," which provides a distinct barrier to the development of an ontology. One of the earliest relevant definitions was offered by Engelbart (1962) when they defined a framework for amplifying human intelligence through assisted decision making and actions. Since then, CAs have been referred to as intelligent agents, intelligent personal assistants, virtual assistants, conversational agents, and software agents. Some researchers define CAs based on their technical capabilities (NSF, 2015; Herron, 2017). For example, Herron defined intelligent agents as "a form of artificial intelligence that operate on user commands" (p. 139). Other researchers have defined agents based on their ability to provide services, information, or work to humans (Santos and Rodrigues, 2013; Lodhi *et al.*, 2018), for example, Lodhi defined virtual assistants as "software that can provide required information or perform work for human(s)" (p. 1). These definitions are broad and can thus be applied to a wide range of different software-based artifacts. For instance, type-ahead-search behaviour is used by a variety of companies (most notably, Google) to autocomplete users search text with the most likely following text. Type-ahead-search behaviour is a form of artificial intelligence that autocompletes based on user commands and performs work for the human user. However, this type of software varies greatly from most common CA archetypes, such as the Amazon Alexa.

Other groups have identified the core components of a CA. The NSF Workshop on Intelligent CAs in 2016 list the following elements of CAs: enhance human capabilities, adapt to dynamic environments,

https://doi.org/10.1017/dsi.2019.270 Published online by Cambridge University Press

cultivate trust within the human-machine relationship, interact naturally, and include multi-disciplinary views. Many of the definitions and descriptions above share similar trends. For example, natural interaction, collaborating, communicative ability, and sociality are all synonymous. Similarly, autonomy/planning and learning/adapting are shared themes throughout the literature. Finally, there is a strong pattern of intelligence-based components, such as, intelligence, reasoning, pro-action, decision-making, problem solving, and proactivity. These four components (learning, intelligence, autonomy, and communication) are widely agreed upon throughout the literature. Therefore, these components will be leveraged to generate the class structure of the ontology proposed here. Table 1 offers a graphical representation of these trends and links to literature.

Table 1. Patterns in component definitions

Author	Intelligence	Autonomy	Learning	Communication	Enhance	Responsiveness
(NSF, 2015)			X	X	X	
(Liu, 2008)	X	X	X	X		
(Reddy, 2000)	X	X	X	X		X
(Li, 2009)	X	X	X			
(Russell and Norvig, 2015)	X	X				
(Saghir and Crespi, 2007)	X	X	X			

In the development of a CA ontology, we subscribe to Le and Watschinski's (2018) definition, that "cognitive assistants offer computational capabilities typically based on Natural Language Processing, Machine Learning, as well as reasoning chains operating on large amounts of data, enabling them to assist humans in cognitive processes" (p. 45). This definition identifies the most common techniques used to accomplish the previously identified components and adheres to the axiom that CAs should enhance and not replace human capabilities. This definition was used to identify the scope and requirements of the ontology.

3 CLASSES AND SUBCLASSES

Four classes were identified in the previous section from a review of the literature: learning (White, 1973; Kazanas and Chawhan, 1975; Gatti-Petito *et al.*, 2013), intelligence (Roberts, 1976; Schwark, 2015; Goksu and Gulcu, 2016; Nkhoma *et al.*, 2017), autonomy (Dorais *et al.*, 1999; Goodrich *et al.*, 2001; Wiest, 2012), and communication (Bateman *et al.*, 1990; Jäger and Rogers, 2012). We have chosen to base the learning subclasses on Gagne's Hierarchical Theory of Learning (Soulsby, 2006), a common hierarchy for learning. Many strong parallels can be drawn between how humans and computers learn using Gagne's Hierarchy (for example, forward and backward chaining). Bloom's Taxonomy (Anderson *et al.*, 2001) is a well cited and commonly used structure for intelligence and is employed here to provide a schema for defining the intelligence subclasses. The Society of Automotive Engineers' Five Level Classification Scale for Autonomous Vehicles (NHTSA, no date) is a current standard for defining autonomous vehicles and was generalized to apply to autonomous assistants. Finally, Richard's Hierarchy of Natural Language Processing Skills (Richard, 2001) is an existing hierarchy for natural language interfaces and was used to define communication subclasses. The following sections describe these classes and their subclasses in further detail.

3.1 Learning

Artificial neural networks, discriminative models, and machine learning techniques (such as deep learning) have improved the ability of CAs to learn about user's intentions. New features such as image recognition and improved natural language processing are possible with these learning techniques. However, not all CAs learn at the same level. Classifying these distinctions will make it possible to explicitly state specific assumptions. The proposed subclasses of learning are based on Gagne's Hierarchical Theory of Learning (Soulsby, 2006). Gagne's Hierarchy has the following levels: signal learning, stimulus-response learning, chaining, verbal association, discrimination learning, concept learning, principle learning, and problem solving. The adapted hierarchy for CAs consolidates signal learning and stimulus-response learning into stimulus learning and removes verbal association.

https://doi.org/10.1017/dsi.2019.270 Published online by Cambridge University Press

The lowest levels of Gagne's Hierarchy are signal learning and stimulus-response learning. Both refer to learning a single function for transforming a single input. Signal learning is the ability to learn to respond to a single signal. A classic example of signal learning is Pavlov's dog or "classical conditioning". Stimulus-response learning is often also referred to as operant conditioning. The primary difference between classical and operant conditioning is the timing of the reinforcing agent (i.e. the reward or punishment). In classical conditioning, the reinforcing agent is presented with the unconditioned stimulus. In operant conditioning, the reinforcing agent is presented after the unconditioned stimulus. This differentiation is not applicable in CAs. Verbal association, while part of Gagne's original hierarchy, is omitted from the set of subclasses because it is assumed that the CA relies on a voice-user interface and requires verbal association at all levels of learning. CAs using machine learning fall within the discrimination and concept learning levels. Table 2 defines the learning levels and provides an example of how it might apply to a CA.

Table 2. Learning hierarchy

Level	Definition	Example
Reinforcement Learning	Respond to a single signal	The verbal command "open", opens the disk tray
Chaining	Connect two or more learned stimulus bonds in a linked sequence	The verbal command "play the morning news", where the "open news app", "search", and "filter" functions are chained
Discrimination Learning	Respond accordingly to a set of similar stimuli that differ systematically	Recognizing different voices of different users
Concept Learning	Respond consistently to a common class of stimuli	Recognizing object classes through image recognition
Principle Learning	Understand the relationship between two or more stimuli and apply this relationship to new or different situations	A trained chatbot uses known question-answer pairs to assess and answer new questions
Problem Solving	Develop unique rules or procedures to solve a problem	A diverse set of heuristic rules are evaluated, down-selected, and used to solve unique tasks

3.2 Intelligence

While Learning classifies how CAs parse inputs, we use the concept of intelligence in our framework to classify how CA's construct outputs. Bloom's Taxonomy is a common hierarchy that has been employed in learning and intelligence for decades. Since CAs are often expected to display human-type intelligence, Bloom's Taxonomy is well-suited for described a CA's intellectual output. Prior studies have tied Bloom's Taxonomy to Web Based Expert Systems and emotionally intelligent machines (Schwark, 2015; Goksu and Gulcu, 2016). Bloom's Taxonomy has the following levels: remember, understand, apply, analyse, evaluate, and create (Anderson *et al.*, 2001).

Table 3. Intelligence hierarchy

Level	Definition	Example
Remember	Recognition and recall of information	A search function of a document
Understand	Interpret, classify, summarize, and infer	Translating a written equation into a symbolic equation
Apply	Understand information and execute based on this understanding	Performing a series of appropriate calculations on a dataset based on an understanding of the context
Analyze	Break down a unit into its component parts, as to understand the organizing structure	Gathering information from a program to decipher the underlying cause of an error
Evaluate	Exercise judgement of a unit	Selecting an effective solution method to a system error
Create	Generation of a unique idea	Designing a novel feature to improve the effectiveness of a system

3.3 Autonomy

Autonomy refers to the ability of the assistant to decide on the sequence of actions required to achieve the user's requested task (Reddy, 2000). The Society of Automotive Engineers (SAE) have developed a 5-Level Classification Scale for Autonomous Vehicles (NHTSA, no date) that has been modified here to apply to autonomous virtual assistants. While the SAE's scale identifies specific levels based on the automation of braking, steering, and the combination thereof, these definitions have been generalized to

https://doi.org/10.1017/dsi.2019.270 Published online by Cambridge University Press

functions, subsystems, and systems. This change allows for a general classification scale that can be used for any automated assistant. However, because of these broad classifiers, autonomy is context dependent. What constitutes a function, subsystem, and system are defined by the user. For example, within Computer-Aided Design (CAD) systems, part generation could be considered the system. However, this system could be expanded with part generation, part assembly, and assembly drawings as subsystems in a broader system. Table 4 defines the five levels of the autonomous classification scale for CAs and provides examples in relation to a CAD system (for example, Inventor or SolidWorks).

Table 4. 5-Level classification scale for CAs

Level	Definition	Example (for a CAD system)
None	No automation	Not applicable
Single Function	A single function is automated	The extrusion function
Subsystem	A single subsystem is automated	Automatic part generation
Conditional (Attention Required)	The system is automated under specific circumstances. The system requires user monitoring	Parts are automatically generated, assembled, and drawings are produced but only for specific part geometries and assembly libraries; the process requires user monitoring
Conditional	The system is automated under specific circumstances. The system does not need to be monitored during automation	Parts are automatically generated, assembled, and drawings are produced but only for specific part geometries and assembly libraries
Full	Full autonomous under all circumstances	Parts are automatically generated, assembled, and drawings are produced

3.4 Communication

CAs communicate with users through natural language processing and voice-user interfaces. Since GUIs are often not used with CAs, the ability to communicate verbally is crucial to a CA's success. Richard (2001) defines the Hierarchy of Natural Language Processing Skills: labelling, functions, associations, categorization, similarities and differences, multiple meanings, idioms, and analogies. Labelling, the naming of objects, is the lowest level of the Hierarchy of Natural Language Processing Skills. It is the vocabulary of the CA. Once an object has a name, a function can be applied to that object. This is defined in the functions level. In the associations level, the CA can associate an object with other objects based on similar or corresponding functions. The categorization level of communication allows CAs to group objects. To group objects, the properties, functions, and associations of objects must first be understood.
Once objects have been grouped, similarities and differences of grouped objects can be identified through distinguishing features. The similarities and differences levels allow for the ability to compare and contrast. Through the multiple meanings level, CAs can understand multiple meanings of the same word based on context and associations. In the idioms level, CAs can understand groups of words whose meaning is established through context rather than deducible by their individual meanings. In the analogies level, the CA can understand the relationship between different parts of language and describe the relationship between multiple concepts.

Table 5. Hierarchy of natural language processing skills

Level	Definition	Example
Labelling	Words are associated with object properties	"Smile" is labelled as happy to identify emotions
Functions	Words and word properties are associated with system functions	Specific commands, such as "open" are associated with system functions
Associations	Like words are identified based on object functions and properties	"Create" and "make" are associated
Categorization	Categories are understood based on word associations	"Guitar" and "trumpet" are categorized as instruments
Similarities and Differences	Distinguishing features are identified	"Mug" is distinguished from "cup" based on the handle
Multiple Meanings	Context is understood to distinguish between multiple meanings of words	"Wind" and its multiple meanings are defined based on context
Idioms	Context is understood to deduce word meaning rather than through individual meanings	"Speak of the devil" is defined based on context
Analogies	The relationship between multiple concepts can be understood	"That's as useful as rearranging deck chairs on the Titanic" links the concepts of being useful and the Titanic

https://doi.org/10.1017/dsi.2019.270 Published online by Cambridge University Press

4 CASE STUDIES: CLASSIFICATION OF COGNITIVE ASSISTANTS

Several institutions have developed CAs to solve problems in fields ranging from education to health care. In this section, three assistants are described and classified using the proposed ontology. The Pennsylvania State University Libraries Database and Google Scholar were queried using the keywords “CA”, “digital assistant”, “intelligent personal assistant”, and “virtual assistant”. 150 articles were analysed, of which 9 CAs were selected based on the definition by Le and Watschinski’s (2018). These 9 assistants and their classifications can be seen in Table 6. Three of the assistants (StuA, Sirius, and Daphne) are explored in further details in the following sections.

Table 6. CA classifications

Name	Learning	Intelligence	Autonomy	Communication	Reference
StuA	Chaining	Remember	Single Function	Function	(Lodhi *et al.*, 2018)
Sirius	Concept	Apply	N/A	Multiple Meanings	(Hauswald, Laurenzano and Zhang, 2015)
Daphne	Concept	Evaluate	Subsystem	Multiple Meanings	(Bang *et al.*, 2018)
ADVICE	Chaining	Remember	Single Function	Categorization	(Garcia-Serrano, Martinez and Hernandez, 2004)
PExA	Stimulus	Apply	Subsystem	Categorization	(Myers *et al.*, 2007)
EmIR	Concept	Apply	Conditional	Labelling	(Rincon *et al.*, 2018)
Duke	Concept	Apply	Subsystem	Associations	(Coronado *et al.*, 2018)
LIZA	Stimulus	Remember	Single Function	Functions	(Le and Wartschinski, 2018)
Personal Health Management Assistant	Chaining	Understand	Conditional (Attention Required)	Associations	(Ferguson *et al.*, 2010)

4.1 StuA

StuA is a CA developed at the Jaypee Institute of Information Technology in Noida, India. StuA was developed to answer the questions of new students regarding grades, facilities, activities, and more. Like many assistants, StuA’s purpose is to connect novice users with domain expertise without needing a physical domain expert. The model was validated and shown to have a 99% accuracy. StuA uses a graphical interface with constrained natural language inputs. StuA uses forward chaining to answer “what will happen” questions and backward chaining to answer “when will this happen” questions. Forward chaining was implemented through CLIPS (software used for the construction of rule and/or object based expert systems) and backward chaining through a Java extension of CLIPS. Chaining is a reasoning method used in inference engines and is the second level of learning. StuA retrieves information from a database. StuA does not understand, analyse, apply, evaluate, or create information. Thus, StuA is ranked in the “Remember” level of the intelligence class.

StuA provides only a single function through their inference engine, the retrieval of information regarding the student’s question. The system cannot execute commands regarding student inquiries such as “schedule an appointment with my advisor.” Due to this limitation, StuA ranks under the Single Function subclass in the autonomy class. StuA uses a GUI to interact with students. Students create sentence of a fixed structure using drop-down menus and open-ended text fields, further constraining their input. Through these constraints, StuA can interact with its database of labelled information. The drop-down menu allows students to select from question types which associates with a specific function (either direct lookup, forward, or backward chaining). This is the second level of the communication class.

4.2 Sirius

Sirius is a CA developed by Clarity Lab at the University of Michigan. It was developed to test the implications of future large-scale computers on the domain. Sirius is an advanced CA that recognizes natural language and images as input. Additionally, Sirius can respond to both questions and action requests. Sirius’s question-answering service uses regular-expression matching, word stemming, and part-of-speech tagging to classify and understand input. The image matching service uses feature extraction to downsample and find/store image key points. Then, the Feature Descriptor orients key points and matches them to a database using an approximate nearest neighbour search. These

https://doi.org/10.1017/dsi.2019.270 Published online by Cambridge University Press

processes allow Sirius to discriminate between various similar stimuli and recognize common classes of information. For this reason, Sirius is ranked at the Concept Learning level of the learning class.
Sirius can output executable actions provided by the system or answers to questions. However, Sirius is not able to analyse component parts, evaluate units, or create novel solutions. Thus, Sirius is ranked at the Apply level of the intelligence class. Sirius is currently a platform that is not used in a specific application. Because autonomy is context dependent and Sirius's context is undefined at this time, Sirius's autonomy level cannot be specified. Sirius's automatic speech recognition process uses either a Hidden Markov Model (HMM) with a Gaussian Mixture Model or an HMM with a Deep Neural Network. This allows Sirius to understand advanced language patterns. Sirius is ranked in the Multiple Meanings subclass of the communication class.

4.3 Daphne

Daphne was developed at Cornell University in Ithaca, New York. She is an intelligent assistant that aids system architects design Earth-orbiting satellites. Daphne provides useful information, can analyse data, and provides feedback on designs. Architects interact with Daphne through a graphical-user and voice-user interface.
Daphne's Brain has two processes: HTTP/WS requests and a Sentence Processor. The HTTP/WS requests handle users working through the graphical-interface. The Sentence Processor handles users working through the voice-user interface. It classifies the intent of the user through a deep learning model based on a Convolutional Neural Network, a type of feed-forward artificial neural networks. Due to this, Daphne is classified at the Concept Learning level of the learning class. The voice-user interface allows architects to naturally interact with Daphne rather than operating through writing code.
Daphne does not merely answer questions. She is also able to analyse designs through historical information and features shared by multiple design types. Daphne can also criticize proposed designs and indicate the strengths and weaknesses. This is part of the Evaluate level of the intelligence class. Although Daphne can perform these various tasks individually, she is unable to chain them into a fully automated process that designs satellites. This is under the Subsystem classification of the autonomy class. Finally, Daphne's voice recognition system uses the Web Speech Application Program Interface, based on Google's Speech-to-Text system. This is a system that can understand advanced language barriers such as multiple meanings.

5 CONCLUSION

CAs are a promising technology that hold solutions ranging from daily questions to the design of Earth orbiting satellites. They provide designers and engineers with easily accessible knowledge-based systems, computational tools, etc. However, CAs currently lack classification and standardization across the field. As governing bodies begin to regulate this growing technology, this lack of understanding will impede the development of these policies. This may be especially harmful to technologies aimed at industries like healthcare, where vulnerable populations are common and increased regulation is imperative. This paper has sought to take the first steps in addressing this problem by identifying a definition for CAs and defining a class and subclass structure for an ontology of the domain.
Four classes were identified: learning, intelligence, autonomy, and communication. Learning and intelligence were used to parameterize how CAs handle input and output, respectively. Autonomy defined how well CAs can generate and execute plans. Finally, the communication class defined the CAs ability to interact with users through voice-enabled technology. The ontology provides engineers with potential design constraints used to narrow the solution space. It also establishes a common domain knowledge for CA creators and users.
Nine CAs were selected and classified using the ontology. From these classifications, the differences in complexity and purpose become more apparent. Simple question-answer services, such as Stu-A, were classified in the lower subclasses (chaining, remember, single function, and functions) while advanced design aids, like Daphne, were classified in the higher subclasses (concept, evaluation, subsystem, multiple meanings). Further CAs, markedly those that exist in industry, should also be investigated in future work to ensure the viability of the ontological structure across all fields.
While this ontology lays the ground work for the classification of a field that currently lacks standardization, it is not yet complete. Additional classes and subclasses should be explored and expanded. For example, machine learning is often characterized through supervised, unsupervised,

https://doi.org/10.1017/dsi.2019.270 Published online by Cambridge University Press

semi-supervised, and reinforcement learning (Dunjko *et al.*, 2016). Such existing categorizations should be integrated into the proposed CA ontology as a means of linking it to relevant fields. In addition, this ontology provides a structure which can be used to guide ongoing research and development in CA technologies. The utility of the ontology for guiding future development should be evaluated through experimental design studies.

REFERENCES

Anderson, W. L. et al. (2001), "A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, abridged edition".

Bang, H. et al. (2018), "Daphne: An intelligent assistant for architecting earth observing satellite systems", *AIAA Conference Proceedings*. Available at: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-1366.

Bateman, J. A. et al. (1990), "A general organization of knowledge for natural language processing: The penman upper model", Technical report, USC Information Sciences Institute.

Bernard, Z. (2018), "Markets Insider", in, Available at: https://markets.businessinsider.com/news/stocks/smart-speakers-are-taking-off-with-consumers-charts-2018-4-1021723081.

Berry, P. et al. (2017), "Evaluating intelligent knowledge systems: experiences with a user-adaptive assistant agent", *Knowledge and Information Systems*. Springer London, Vol. 52 No. 2, pp. 379–409. http://doi.org/10.1007/s10115-016-1011-3.

Boost (2018), "boost.ai", in. Available at: https://www.boost.ai/technology/.

Buchanan, B., Sutherland, G. and Feigenbaum, A. E. (1966), "Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry", Available at: https://profiles.nlm.nih.gov/ps/access/BBABKI.ocr.

Bush, V. (1966), "Memex revisited", Science Is Not Enough.

Champin, P.-A. (2003), "ARDECO: an assistant for experience reuse in Computer Aided Design", *Workshop from Structured Cases to Unstructured Problem Solving Episodes - WS 5 of ICCBR'03*, Trondheim (NO), pp. 287–294. Available at: http://liris.cnrs.fr/~pchampin/publis/iccbr2003c.pdf.

Chang, X. and Rai, R. (2010), "Development and utilization of ontologies in design for manufacturing", *Journal of Mechanical Design*, Vol. 132 No. 2. Available at: http://computingengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1472631.

Coronado, M. et al. (2018), "A cognitive assistant for learning java featuring social dialogue", *International Journal of Human-Computer Studies*, Vol. 117. http://doi.org/10.1016/j.ijhcs.2018.02.004.

Dorais, G. et al. (1999), "Adjustable autonomy for human-centered autonomous systems on mars", *Working notes of the Sixteenth International Joint Conference on Artificial Intelligence Workshop on Adjustable Autonomy Systems*, pp. 16–35.

Dunjko, V. et al. (2016), "Quantum-enhanced machine learning", Available at: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.117.130501.

Engelbart, D. C. (1962), "Augmenting human intellect: A conceptual framework", Available at: https://www.dougengelbart.org/pubs/augment-3906.html (Accessed: 23 July 2018).

Ferguson, G. et al. (2010), "Towards a personal health management assistant", *Journal of biomedical informatics*, Vol. 43(5 Suppl), pp. S13–S16. http://doi.org/10.1016/j.jbi.2010.05.014.

Garcia-Serrano, M. A., Martinez, P. and Hernandez, Z. J. (2004), "Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce", *Expert systems with applications*, Vol. 26 No. 3, pp. 413–426. Available at: https://www.sciencedirect.com/science/article/pii/S0957417403001726.

Gatti-Petito, J. et al. (2013), "Clinical scholarship and adult learning theory: A role for the DNP in nursing education", *Nursing Education Perspectives*, Vol. 34 No. 4. Available at: https://journals.lww.com/neponline/Fulltext/2013/07000/Clinical_Scholarship_and_Adult_Learning_Theory__A.14.aspx.

Goksu, I. and Gulcu, A. (2016), "The Evaluation of the cognitive learning process of the renewed bloom taxonomy using a web based expert system", *TOJET: The Turkish Online Journal of Educational Technology*, Vol. 15 No. 4.

Goodrich, A. M. et al. (2001), "Experiments in adjustable autonomy", *Proceedings of IJCAI Workshop on autonomy, delegation and control: interacting with intelligent agents*. American Association for Artificial Intelligence Press.

Hatchuel, A. and Weil, B. (2009), "C-K design theory: An advanced formulation", *Research in Engineering Design*, Vol. 19 No. 4, pp. 181–192. http://doi.org/10.1007/s00163-008-0043-4.

Hauswald, J., Laurenzano, A. M. and Zhang, Y. (2015), "Sirius: An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers", *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 223–238. Available at: https://dl.acm.org/citation.cfm?id=2694347.

Herron, J. (2017), “Intelligent Agents for the Library”, *Journal of Electronic Resources in Medical Libraries*. Routledge, Vol. 14 No. 3–4, pp. 139–144. http://doi.org/10.1080/15424065.2017.1367633.
Jäger, G. and Rogers, J. (2012), “Formal language theory: refining the Chomsky hierarchy”, *Philosophical Transactions of the Royal Society B*, Vol. 367 No. 1598.
Kazanas, C. H. and Chawhan, R. A. (1975), “Effects of two treatments on cognitive achievement of students varying in problem-solving ability”, *The Journal of Educational Research*, Vol. 68 No. 7, pp. 269–273.
Kumar, R. and Rasal, A. (2018), “Smart speaker market by intelligent virtual assistant (amazon alexa, google assistant, siri, cortana, and others), end user (personal and commercial), distribution channel (online and offline), and price (low, mid, and premium) - global opportunity analys”, *Allied Market Research*.
Kuznar, D. et al. (2014), “Virtual assistant platform”, *An International Journal of Computing and Informatics*, Vol. 40 No. 3. Available at: http://www.informatica.si/ojs-2.4.3/index.php/informatica/article/view/1437.
Le, N.-T. and Wartschinski, L. (2018), “A cognitive assistant for improving human reasoning skills”, *International Journal of Human-Computer Studies*, Vol. 117. http://doi.org/10.1016/j.ijhcs.2018.02.005.
Li, F. (2009), “An intelligent assistant design system based on CBR and MAS”, *2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design. IEEE*. Available at: https://ieeexplore.ieee.org/abstract/document/5375257/.
Liu, G. (2008), “The application of intelligent agents in libraries: a survey”, *Program*, Vol. 45 No. 1, pp. 78–97. Available at: https://www.emeraldinsight.com/doi/abs/10.1108/00330331111107411.
Lodhi, P. et al. (2018), “StuA: An Intelligent Student Assistant”, *IJIMAI Journal*, pp. 1–9.
Myers, K. et al. (2007), “An intelligent personal assistant for task and time management”, *AI Magazine*, Vol. 28 No. 2, pp. 47–62. http://doi.org/10.1609/aimag.v28i2.2039.
NHTSA (no date), “United States Department of Transportation”, in.
Nkhoma, Z. M. et al. (2017), “Unpacking the revised Bloom’s taxonomy: developing case-based learning activities”, *Education & Training*, Vol. 59 No. 3.
NSF (2015), “Intelligent cognitive assistants”.
Perez, S. (2018), “Tech crunch”, in. Available at: https://techcrunch.com/2017/07/11/siri-usage-and-engagement-dropped-since-last-year-as-alexa-and-cortana-grew/.
Reddy, M. P. (2000), “Mobile agents”, *Resonance*, Vol. 7 No. 2. Available at: https://link.springer.com/article/10.1007/BF02836751.
Richard, J. G. (2001), *The Source for Processing Disorders*. LinguiSystems, Inc.
Rincon, J. et al. (2018), “A new emotional robot assistant that facilitates human interaction and persuasion”, *Knowledge and Information Systems*. Springer London, pp. 1–21. http://doi.org/10.1007/s10115-018-1231-9.
Roberts, N. (1976), “Further verification of bloom’s taxonomy”, *The Journal of Experimental Education*, Vol. 45 No. 1, pp. 16–19.
Russell, J. S. and Norvig, P. (2015), “Artificial intelligence: a modern approach”, *Pearson*. Available at: http://thuvien.thanglong.edu.vn:8081/dspace/handle/DHTL_123456789/4010.
Saghir, B. and Crespi, N. (2007), “An intelligent assistant for context-aware adaptation of personal communications”, *2007 IEEE Wireless Communications and Networking Conference*. IEEE. Available at: https://ieeexplore.ieee.org/iel5/4204175/4204176/04204648.pdf.
Samuel, L. A. (1959), “Some studies in machine learning using the game of checkers”, *IBM Journal*, pp. 535–554.
Santos, J. and Rodrigues, C. J. J. P. (2013), “Intelligent personal assistants based on internet of things approaches”, *IEEE Systems Journal*, Vol. 12 No 2. Available at: https://ieeexplore.ieee.org/abstract/document/7473837/.
Schwark, D. J. (2015), “Toward a taxonomy of affective computing”, *International Journal of Human-Computer Interaction*, Vol. 31 No. 11, pp. 761–768.
Shen, W., Norrie, D. H. and Barthès, J.-P. A. (2010), “Multi-agent systems for concurrent intelligent design and manufacturing”, *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*. http://doi.org/10.4324/9780203305607.
Shortliffe, H.E. (1975), “MYCIN: Computer-based medical consultations”.
Soulsby, D. (2006), “Gagne’s hierarchical theory of learning: Some conceptual difficulties”, *The Journal of Curriculum Studies*, pp. 122–132. Available at: https://www.tandfonline.com/doi/pdf/10.1080/0022027750070204.
Weizenbaum, J. (1966), “ELIZA-computer program for the study of natural language communication between man and machine”, *Communications of the ACM*, Vol. 9 No. 1, pp. 36–45. Available at: https://dl.acm.org/citation.cfm?id=365168.
White, T. R. (1973), “Research into learning hierarchies”, *Review of Educational Research*, Vol. 43 No. 3, pp. 361–375. Available at: http://journals.sagepub.com/doi/pdf/10.3102/00346543043003361.
Wiest, G. (2012), “Neural and mental hierarchies”, *Frontiers in Psychology*, Vol. 3.
Wu, S. et al. (2006), “Personal assistant agents for collaborative design environments”, *Computers in Industry*. http://doi.org/10.1016/j.compind.2006.04.010.