ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 8 , August 2017

An Intuitive Architecture for Next Generation Digital Personal Assistants

[1] Mehul Jain, [2] Utkarsh Bhatia
[1][2] Indian Institute of Technology (ISM), Dhanbad

***Abstract*— Current voice-based digital assistants despite their claims of being intelligent, lack abilities that a true personal assistant must possess like extendable skill set, dynamic adaptation and high context awareness. In this paper, we highlight some design and implementation requirements that must be met in order for the development of next generation digital personal assistants and propose a general architectural backbone that can be used to make headway for such personalized speech-operated assistive technology. In particular, we confer about issues of extensibility of the skill set used by the digital assistant, hypothesis generation and evaluation, extensive user adaptation, and redundant representations handling in the design. Further, we briefly discuss the research and development directions that are undertaken to tackle challenges put by such a system. We then consider a scenario and illustrate the data flow in our architecture.**

***Keywords*— General Architecture, Digital Personal Assistants, Extensible Skill Set, Dynamic Adaptation, Context Awareness.**

INTRODUCTION

Digital personal assistants (DPA) are playing an integral role in day-to-day activities of many people. Such systems can easily be employed to augment the capabilities of a user by handling a barrage of routine tasks, such as keeping up with their emails and keeping their calendars up-to-date. Email, calendaring systems, daily planners, and web searches are now easily managed by these systems like Google now [1], Microsoft's Cortana [2] and Apple's Siri [3].

Although today's system provides a limited set of skills and does not allow us to append to the list, individual applications are getting more and more sophisticated providing extensive and rich application features. Users can be provided a way to map their high-level goals to the low-level vocabulary of specific applications and services using special programs or "task assistants" to reach a given goal [5].

When it comes to personal assistants it's important for them to adapt to the task at hand, habits, and interests of the user to continuously improve the user's experience by inferring user's intent, making relevant suggestions and maintaining context. Apart from this, adapting to the language, emotions, and personality of the user is also important for a true personal assistant.

Such systems may benefit greatly by using a user model and context awareness to generate hypothesis and make inferences. For example, when the user asks the assistant about going to some city, the system should be able to come up with a hypothesis and recommendations like "the user may need to book a flight" or that "the user may need to book a hotel". The hypothesis is then evaluated by soliciting a response by the user and then analyzing it that improves the user model and hence helps in generating more apt hypothesis in future.

We put forward a general multi-agent architecture for intelligent context-aware system that tries to incorporate these details for the implementation of DPA using the marriage of speech recognition with advanced natural language processing techniques, scalable inferences, semantic technologies and user modeling. The quality of proactive decisions is improved in our architecture by virtual agents by enabling them to seek explanations, make predictions, generate and test the hypothesis and perform what-if analysis.

Our architecture also provides a provision for adding and removing new skills anytime. Moreover learning is at the core of the architecture: over the time system adapts to the need and preferences of the user. We allow our architecture to handle redundancy at various levels so as to keep multiple representations and choices available at all the time and hence reducing errors.

In this paper, we have focused on highlighting some design and implementation requirements that must be met in order for the development of next generation digital personal assistants and have proposed a general architectural backbone that can be used to make headway for such personalized speech-operated assistive technology. In particular, we confer about issues of extensibility of the skill set used by the digital assistant, hypothesis generation and evaluation, extensive user adaptation, and redundant representations handling in the design. Further, we briefly discuss the research and development directions that are undertaken to tackle challenges put by such a system.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 8, August 2017

II. DESGIN REQUIREMENTS

One perspective of looking at the system is the quality of services it provides to the user. In the case of a DPA, the quality can be seen tantamount to the extent to which the system can complement what the user normally do [5]. The following requirements are such that they help in enhancing the overall quality of the DPA:

1) Extensibility: It should be feasible to augment the ability of the system. For example, it should be feasible to program a task assistant and if a new task agent is generated it should be easily pluggable in the system without disrupting the old setting and dovetail with the existing assistance provided. It will allow the user to add and remove capabilities from the system on the fly.

2) Adaptability: The system should be able to conform with the user by adapting to the user's actions and habits, not just to in-session tasks. It should be able to make inferences about the user's action based on how they usually carry out their task, so as to improve the quality of proactive decisions made by the system and perform appropriate task. Having knowledge about the user also helps in improving the context awareness which in turn improves the proactive decision-making [4].

3) Hypothesis Generation and Evaluation: The system must also be provided a way to make hypothesis and prediction about the user. Evaluation of these hypothesizes and predictions empower us to better model the user. We can further improve the feature if we allow the virtual agent to seek an explanation in a way which can be used in future for enhanced decision making. Performing what-if analysis helps the assistant to consider different scenarios, then by interacting with the user to solicit a response for these what-if queries the system can augment the user model.

Apart from this, the design should be such that it mitigates error at each level so that they do not accentuate in the next processing step. Implementing the systems with robust knowledge base can help us in applying consistency check at every major step of our application. Knowledge base plays an important role in the implementation of many modules. Automatic speech recognition module uses dictionary and language models for recognizing speech; Language understanding module uses lexical, syntactic and semantic knowledge for extracting the meaning of a sentence; user models are used for making inferences; task hierarchies and domain ontology are maintained for task management. Communication between modules and knowledge base should be efficient and reliable so that required knowledge can be made available to the respective module at any time. Implementing sound and complete reasoning and inference also allows us to mitigate error at each step.

In addition to these requirements, there are a number of other more standard system-oriented qualities from the engineering perspective, such as robustness, availability, security and efficiency. All these qualities should be kept in mind while designing a DPA so that these systems can be of use to the user in unprecedented ways.

III. ARCHITECTURE

We build on previous long-term research on HALO [6] at Vulcan Inc., CALO [7] led by SRI (part of which was spun-off as Siri [8] later popularized by Apple Inc.), Cyc [9],[10] and Whodini [11].

To achieve the goals we have adopted modular and centrally controlled run-time architecture, pictured in figure 1, which portrays the architecture from a single user perspective. We first talk about all the components and how they fit in the bigger picture, then we discuss each component in turn.

The first piece of our system is an automatic speech recognizer which converts the user's speech to text. The produced text is then processed through a series of lexical, syntactic and semantic analysis. Then anaphora resolution, ellipsis handling, and reference resolution are tackled to generate a complete meaning representation of what is said by the user. This complete meaning representation is then passed on to a dialogue Manager. Dialogue Manager is responsible for planning and organizing the actions of the system. It manages communication with all the modules and decides which action is to be invoked and also decide how interaction with the user will be handled. Discourse Manager is then invoked to put this new user information into the context in the working memory and it tells us where in the discourse we are. Inference engine tries to find as many inferences from the updated working memory then incorporates the inferences back into the memory. New hypotheses are generated based on the working memory and user models. Discourse Manager, inference engine, and hypothesis engine are invoked every time new information comes into the system i.e. either by the user or the task specialist. Activity Manager is asked by the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 8, August 2017

dialogue Manager to invoke a specialist for some task that is to be accomplished. Task specialists are simply programs which communicate with the applications to realize the task at hand. The result of the task and the hypothesis generated are sent back to the Dialogue Manager which decides how the result will be conveyed back to the user and hypothesis will be posed to the user for evaluation. The Dialogue Manager returns a computer representation of its decision. This computer representation is converted back into natural language text which in turn is converted into speech.

1) Working Memory (Blackboard): At initialization, this memory consists of user models and a general profile. As the discourse begins it maintains:

- Discourse history which is the short term memory of the system i.e. the current context.
- Activity Working Memory which takes care of storing all information about current activity and previous activities performed.
- Working Memory also stores internal procedural information and information that keeps track of the current state of the system and all decision taken by the system so far.

User's words, co-occurrence, and the temporal sequence of queries are used to maintain the discourse history i.e. the current context [12], [13].

2) Automatic Speech Recognizer: Goal of this module is to receive the user's speech and generate as output sequence of words that most likely correspond to what the user said. One of the most prevalent approaches is a stochastic method based on acoustic and language models [14]. Usually, the set of all the words considered by these models is stored in the so-called dictionary. However, recent approaches to ASR employ wide coverage models that do not require a dictionary. For example, the approach used in the Google speech API employs a knowledge graph. The result of ASR is processed to assign a confidence score to each word, which depicts its probability of being correctly recognized. If this score falls below a minimum threshold the user is asked to provide the information again. There are a number of factors which make this task challenging like, environmental conditions (e.g. noise, etc.), the acoustic similarity between words, and phenomena concerned with spontaneous speech, such as false starts, filled pauses, and hesitations

3) Language Understanding Module: The goal of this module is to obtain the semantic representation of the input in form of one or more frames. The text is passed through a series of lexical, syntactic and semantic analysis. The task to be performed by the module is challenging due to the specific difficulties inherent in the processing of natural languages, such as ambiguity, anaphora, and ellipsis. To carry out language understanding, this module typically employs grammar rules or statistical approaches, or some combination of both [15].

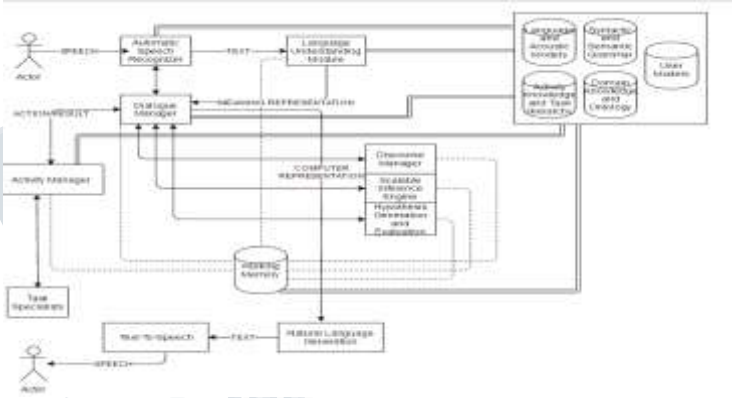

Figure 1

4) Dialogue Manager: Dialogue Manager controls a major part of the application. It is this module's job to decide which action will be invoked. The work of the dialogue manager is inherently context-dependent; it uses working memory to take all the decisions. After a task has been performed and a hypothesis is generated the results are passed to the dialogue manager which then conveys the result to the user. Dialogue Management is based on either stochastic processes i.e. MDPs [16], POMDPs [17], the information state principles [18], or a hierarchy of tasks [19, 20]. Each of these categories has advantages and drawbacks. The information state paradigm is considered as an inaccurate theoretical framework and therefore does not meet the requirements of most practical implementations. The same applies to stochastic processes whose main drawback is that training data has to be collected in order to build a system.

5) Discourse Manager: This module uses the working memory to keep the status of the discourse and is

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 8, August 2017

responsible for telling the dialogue manager where in the discourse we are. For e.g. if the user shows an intent for something we should be able to tell that it's our turn to seek information.

6) Inference and Hypothesis engine: This part of the architecture is important as it help in proactive decision making. Inferences and hypothesis generated helps in collecting information that may be valuable for decision making in future. This part is responsible for finding entailments, inherent assumptions and implications of a given utterance by the user.
These tasks use the working memory to make inferences. Hypotheses are generated by first finding related entities and relationships of the action or object and then possible attributes are assumed for those entities and relationships using the user model. The generated hypothesis is posed to the user and response is used to evaluate the same. Methods like collaborative filtering can be used for hypothesis generation [4].

7) Activity Manager: As soon as discourse manager asks activity manager to perform a task, the module obtains activity specific information from the domain ontology, identifies the information need (if any) and identifies how to prioritize the information needed. For e.g. to book a flight it would be useful to filter on time first and then money. Then the requirements are sent to the dialogue manager. When the activity manager receives all the required information it checks whether the provided information enables us to perform the activity. It updates the activity working memory to maintain the current activity status of the system. This module is responsible for adding and making changes in ontology, information needs hierarchy and activity knowledge base every time a new skill is added. The challenges in this module are due to uncertainties, missing ontology, and corrupt information.

8) Natural Language Generation Module: The decision of dialogue manager is an abstract computer representation which has to be converted into grammatically and semantically correct natural language. To be coherent with the current status already conveyed information is omitted (ellipsis) and nouns are replaced by pronouns (anaphora). It's also taken care that redundant information is not conveyed, for e.g. the information which can easily be inferred should be omitted. Template-based methods [21] or more recent and sophisticated approaches such as statistical [22], [23] are used.

9) Text to Speech Module: Finally, the sentences in text format generated by the NLG module are the input to the last module shown in Figure 1. This task is difficult due to reasons such as abbreviations in the sentence, number and different pronunciation (interrogative vs. declarative). To do this task, first numbers and abbreviations are replaced by corresponding words and then words are marked which indicates how the word should be pronounced (for e.g. intonation and speed). As opposed to other simple methods for speech synthesis based on concatenation of pre-recorded words, the TTS process [24] allows transforming into speech any arbitrary text.

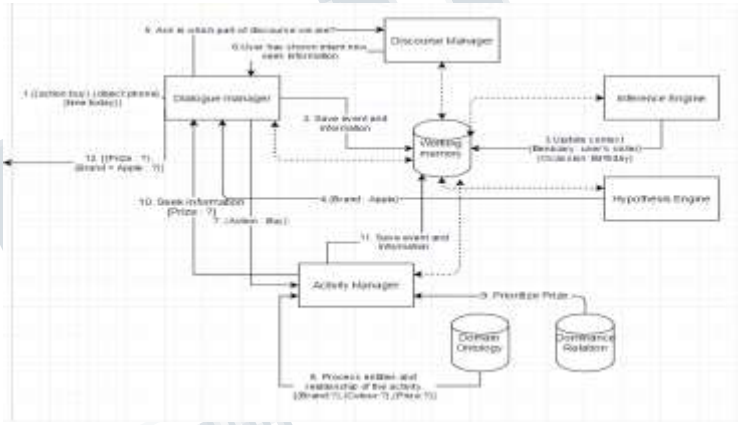

Figure 2

IV. SCENARIO

Consider the following scenario:
You saved a note to buy a new phone as a birthday gift for your sister on DATE: xx/yy/zzzz. Along with this we have the world knowledge i.e. this user owns many apple devices.
Following the same on xx/yy/zzzz you utter to your DPA "I want to buy a new phone". We show how the following string, after we have obtained a semantic representation, is processed through our architecture and highlight the data flow in figure 2.

V. USER MODELING

With the advances in DPA, there is an excellent opportunity to build richer user models. At the beginning, the capabilities were limited to very small vocabularies and so the application development was much easier. With current breakthroughs, technology became more sophisticated and study started on how to tackle such

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 8, August 2017

large information set and vocabulary to enhance user experience. Now the user is the center of the system design, instead of the application domain.

Nowadays, the information about the user is not only considered at design time but also included in modules that allow the system to dynamically adapt to the user's state. It is possible to obtain knowledge about not only what the user say, but also how they say it, where they say it and even predict why they said it and what they will say next and these abilities will be increasingly more sophisticated in the future.

Modeling at different levels is often found useful while designing such systems. In the speech recognition step the language model can be trained to adapt the user. Emotions of the user can be modeled and used either as a new source of information [25] or to build more social relations to sustain more complex forms of affect such as engagement and trust [26]. The system may include complex models on how emotions vary over time. Not only context and emotion determine our behavior, they are also modulated by our personality [27]. Nass and Yen [28] showed that if the user perceives the agent's personality as its own the users' perception of the system's intelligence and competence increases. User's habits, interests [4] and actions can be modeled to increasing user experience and also helps in making a better hypothesis. Contextual models are very important for DPAs due to various reasons. Firstly, it allows obtaining a better system performance; for example, it is possible to use different noise models that allow increasing the speech recognition rates. Secondly, the location information along with the general profile of the user and the interaction can be used to deliver functionalities such as find nearby spots or to recognize the activities being carried out by the user to provide adequate services [29].

VI. CONCLUSION

While implementing this architecture we were able to fulfill the requirements like extending the skill set of the DPA (to include actions like sending gifts, transferring money and booking cabs) using specific task agents and updating task hierarchy and domain ontology according to the actions included. We were also able to work with an improved context (in-session and cross-session) thus generating better hypothesis and recommendations. The evaluation of hypothesis and recommendations helped us to better refine our user-models. Operating as an on-stack application we were able to use other application feature and save ourselves from the extra effort of re-engineering the system. Furthermore with development of individual modules and more sophisticated techniques this model can be used for the development of next-generation digital personal assistants.

VII. REFRENCES

[1] Google Now, http://www.google.com/landing/now/.

[2] Microsoft Cortana, http://www.windowsphone.com/en-us/how-to/wp8/cortana/meet-cortana.

[3] Apple Siri, http://www.apple.com/in/ios/siri/

[4] Ramanthan Guha, Vineet Gupta, Vivek Ragunathan and Ramakrishna Srikant. (Feb 2015). User Modeling for a Personal Assistant. WSDM'15, Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.

[5] David Garlan, Bradley Schmerl. (July 2006). Architecture for Personal Cognitive Assistance. Proceedings of the 2006 Conference on Software Engineering and Knowledge Engineering.

[6] Friedland, N. S., P. G. Allen, G. Matthews, M. Witbrock, D. Baxter, J. Curtis, B. Shepard, P. Miraglia, J. Angele and S. Staab. (2004). "Project halo: Towards a digital aristotle." AI magazine 25(4): 29.

[7] Tur, G., A. Stolcke, L. Voss, J. Dowding, B. Favre, R. Fernandez, M. Frampton, M. Frandsen, C. Frederickson and M. Graciarena. (2008). The CALO meeting speech recognition and understanding system. Spoken Language Technology Workshop, 2008. SLT 2008. IEEE.

[8] Gruber, T. (2009). Siri, A Virtual Personal Assistant— Bringing Intelligence to the Interface, Jun.

[9] Panton, K., C. Matuszek, D. Lenat, D. Schneider, M. Witbrock, N. Siegel and B. Shepard. (2006). Common sense reasoning–from Cyc to intelligent assistant. Ambient Intelligence in Everyday Life, Springer: 1-31.

[10] Lenat, D., M. Witbrock, D. Baxter, E. Blackstone, C. Deaton, D. Schneider, J. Scott and B. Shepard. (2010). "Harnessing Cyc to Answer Clinical Researchers ' Ad Hoc Queries." AI Magazine 31(3): 13-32.

[11] Mehra, P. (2012). "Context-aware computing: beyond search and location-based services." Internet Computing, IEEE 16(2): 12-16.
[12] L. Li, H. Deng, A. Dong, Y. Chang, and H. Zha. Identifying and labeling search tasks via query-based Hawkes processes. In SIGKDD, 2014.

[13] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Identifying task-based sessions in search engine query logs. In WSDM, 2011.

[14] Huang, X., Acero, A., & Hon, H.-W. (2001). Spoken language processing: A guide to theory, algorithm and system development. Prentice Hall.

[15] Griol, D., Callejas, Z., López-Cózar, R., &Riccardi, G. (2014). A domain-independent statistical methodology for dialog management in spoken dialog systems. Computer Speech and Language, 28(3), 743–768. http://dx.doi.org/10.1016/j.csl.2013.09.002.

[16] E. Levin, R. Pieraccini, and W. Eckert. (1998). "Using Markov decision process for learning dialogue strategies," Proceedings of ICASSP.

[17] J. Henderson and O. Lemon. (2008) "Mixture model POMDPs for efficient handling of uncertainty in dialogue management," Proceedings ACL-HLT, pp. 73–76, 2008

[18] S. Larsson and D. Traum. (2000). "Information state and dialogue management in the TRINDI dialogue move engine toolkit," Natural language engineering.

[19] D. Bohus and A. I. Rudnicky. (2003) "RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda," in Proceedings of EUROSPEECH, 2003.

[20] C. Rich and C. L. Sidner, "Collaborative discourse, engagement and always-on relational agents," in Proceedings of AAAI, 2010.

[21] Baptist, L., &Seneff, S. (2000). GENESIS-II: A versatile system for language generation in conversational system applications. Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP '00), 3, 271–274.

[22] Dethlefs, N., Hastie, H., Cuayáhuitl, H., & Lemon, O. (2013). Conditional random fields for responsive surface realization using global features. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), 1254–1263.

[23] Rieser, V., Lemon, O., & Keizer, S. (2014). Natural language generation as incremental planning under uncertainty: Adaptive information presentation for statistical dialogue systems. IEEE/ACM Transactions on Audio, Speech and Language Processing, 22(5), 979–994. http://dx.doi.org/10.1109/TASL.2014.2315271

[24] Dutoit, T. (1996). An introduction to Text-to-Speech synthesis. Dordrecht: Kluwer Academic.

[25] Callejas, Z., Griol, D., &López-Cózar, R. (2011). Predicting user mental states in spoken dialogue systems. EURASIP Journal on Advances in Signal Processing, 2001, 6. http://dx.doi. org/10.1186/1687-6180-2011-6

[26] Acosta, J. C., & Ward, N. G. (2011). Achieving rapport with turnby-turn, user-responsive emotional coloring. Speech Communication, 53(9–10), 1137–1148. http://dx.doi.org/10.1016/j. specom.2010.11.006

[27] Callejas, Z., López-Cózar, D., Ábalos, N., &Griol, D. (2011). Affective conversational agents: The role of personality and emotion in spoken interactions. In D. Pérez-Marín& I. Pascual-Nieto (Eds.), Conversational agents and natural language interaction: Techniques and effective practices (pp. 203–222). IGI Global. http://dx.doi.org/ 10.4018/978-1-60960-617-6. ch009

[28] Nass, C., & Yen, C. (2012). The man who lied to his laptop: What we can learn about ourselves from our machines. Current Trade.

[29] Zhu, C., Sheng, W. (2011). Motion- and location-based online human daily activity recognition. Pervasive and Mobile Computing, 7(2), 256–269. http://dx.doi.org/10.1016/j.pmcj.2010.11.004