Daphne: An Intelligent Assistant for Architecting Earth Observing Satellite Systems

Hyunseung Bang[*], Antoni Virós[*], Arnau Prat[*], and Daniel Selva[†]
Cornell University, Ithaca, NY, 14850, USA

I. Introduction

Over fifteen years after the publication of foundational work in the field by Rechtin and Maier, system architecting remains an art more than a science:[1] a process driven for the most part by a small team of system architects. Indeed, architecting complex systems is too challenging a task to automate, for a number of reasons: it is a relatively unstructured and ill-defined task that requires creativity and adaptation, and casting the problem as a search/optimization/decision-making problem results in a space of alternatives with intricate and non-linear couplings between variables, high dimensionality of the decision space, and technical and programmatic uncertainty and ambiguity.[2,3] Furthermore, in many cases, the mapping between the design decisions and criteria is a complex blackbox function containing expensive calculations and simulations. As pointed out many times before, humans are better than computers at thinking holistically, and navigating unstructured problem spaces.

However, this does not mean that computational tools should not be used for system architecting. Research in the last 15 years has shown that using an interactive process in which the human is assisted by optimization and visual and data analytics tools can bring rigor, exhaustiveness, and consistency into architecting, and help alleviate some of the well-known human cognitive limitations such as biases in decision-making and low "computational power".

Various interactive visualizations and decision support tools have been developed to help system designers analyze complex models and explore the design space (a.k.a. the tradespace).[4–7] These tools support large-scale analysis with thousands to billions of design alternatives, each defined in a multi-dimensional objective space, including various performance and cost/schedule/risk metrics. For example, various visualization tools allow the users to compare different views of the data (e.g., decision space vs objective space, or 2d slices of the objective space), highlight architectures sharing certain features, and steer the search towards a certain direction in the objective space.[4,5,8,9] Some of these tools utilize unsupervised machine learning algorithms such as manifold learning, feature selection, and clustering to help users visualize solutions in a high-dimensional space.[7,10] To further reduce the cognitive load of system designers, other tools combine visualization with data mining algorithms that extract useful knowledge or insights, often in the form of simple logical rules with an if-then structure, such as "IF any spacecraft in the architecture weighs more than 3,000kg, THEN the architecture is likely to have low cost-efficiency".[11–13] The use of logical rules as data structure for these insights has a long tradition in artificial intelligence, and is motivated by evidence that not only are logical rules easy to understand by humans, but they may also be the way human experts actually solve complex problems.[14] On the algorithmic side, the methods to extract such logical rules include decision trees,[15] Algorithm Quasi-optimal (AQ) learning,[12] and association rule mining.[11,13] Regardless of the choice of algorithm, the formal representation of knowledge provides an opportunity to incorporate intelligence into the tools. New knowledge can be learned either through performing inferences or through learning from examples. Then this new knowledge can be used to adapt to different problems or to different phases of the search process.[16–18]

As decision support tools become more intelligent, researchers are paying more attention to human-agent interaction. For more than a decade, the Intelligent Systems and Human-Agent Interaction fields have

[*]Student, Sibley School of Mechanical and Aerospace Engineering, Student Member AIAA.
[†]Assistant Professor, Sibley School of Mechanical and Aerospace Engineering, Member AIAA.

studied how to facilitate effective collaboration between the human and intelligent agents in many different tasks.[19,20] Various intelligent personal assistants or cognitive assistants[21–23] have been developed to reduce the cognitive load of the humans users in their everyday tasks such as organizing e-mails and scheduling meetings. All intelligent assistants consist of some interface that the user can use to ask information, request tasks, etc. Some utilize verbal natural language interfaces,[23,24] while others implement purely visual and form-based interactions where the type of the user input is determined apriori.[22] The roles of these intelligent assistants may also vary depending on the application domain and the task required from the agent. For example, CAMA is a intelligent assistant system that is intended to increase the situational awareness of pilots. CAMA monitors the pilots behavior autonomously, and compares it to a pre-defined normative model. When a pilot's behavior violates critical safety rules, it signals a warning sign and generates proposals to resolve conflicts.[25] On the other hand, an intelligent assistant called CALO, which performs routine office tasks for its user, takes a much more submissive role.[23] Its actions are triggered by the user, and when there is a misalignment between the user's belief and the agent's belief, it seeks an "advice" from the user for more information. While various forms of intelligent assistants have been developed and tested previously for a variety of tasks, they have not yet been applied to the task of designing the architecture of a complex system. Since, as mentioned earlier, architecting complex systems requires dealing with high-dimensional design spaces and complex couplings between variables, system designers are often put under large cognitive loads. Therefore, there is an unexplored opportunity for developing intelligent assistants that can help system designers reduce their cognitive load and explore the tradespace more effectively.

In this paper, we introduce Daphne, a new intelligent assistant specializing in architecting Earth observing satellite systems. Daphne can take two different roles in the design process: 1) Analyst, answering questions about different aspects of the design problem, and 2) Critic, assessing the strengths and weaknesses of architectures proposed by the user, and suggesting ways to improve them. Daphne constructs its outputs from three different sources of knowledge: 1) a historical database of all Earth observing satellite missions ever launched (and some planned missions), 2) a knowledge base containing domain-specific expert knowledge about how to design an Earth observing mission, and 3) knowledge extracted from data mining the current design space. It also has different modes of interaction, including verbal and visual interfaces (graphical user interface including virtual reality, voice-based and text-based natural language processing), as well as a physical embodiment for non-verbal interaction.

The key contribution of this paper is exploring the possibility and the potential benefits of having an intelligent assistant for system architecting. Specifically, this research describes the overall architecture of Daphne, and investigates which aspects of the system are the most critical in reducing the cognitive load of the system designer. Daphne can take two different roles (Analyst and Critic), draw information from three different sources (historical database, expert knowledge-base, and knowledge extracted from data mining), and communicate with the user through verbal and visual interfaces. As Daphne is the first intelligent assistant that specializes in system architecting, it has not been reported how these different components will influence the design process. Therefore, it is important to first identify whether each component of the system is effective in reducing the cognitive load, and if so, how they influence the system designer's experience. As Daphne has a modular structure, we can test different combinations of the system components and perform qualitative assessment of each approach. For implementation and testing the agent, we focus on a real-world problem of architecting an Earth observing satellite system for operational climate monitoring.[26] It is a challenging system architecting problem that has a large design space and exhibits non-linear couplings between input variables. We provide a detailed use case scenario of how Daphne can be applied and contribute to reducing the cognitive load of the system designer.

The remainder of this paper is organized as follows. Section II provides some background on intelligent assistants. Section III provides an overview of the system. Section IV will provide a use case scenario of utilizing Daphne in architecting an Earth observing satellite system. Finally, Section V will be included in the full version of the paper, and will discuss the limitations of the current implementation and describe the future work.

II. Background

Intelligent personal assistants are intelligent agents that manage and perform tasks on behalf of humans to reduce routine tasks and the cognitive load of the human users.[23] It is important to note that the main goal of developing these systems is not replacing human, but to enhance human capabilities both in performing routine tasks and solving complex problems.[27] Many of the intelligent personal assistants currently developed focus on performing routine everyday tasks such as organizing emails, scheduling meetings, booking conference rooms, etc. For example, MailCat helps the users organize emails into different folders using a natural language classifier that learns and adapts to the users habits.[21] ABDUL is a language and information assistant that can translate between English and Thai, and answer general questions such as weather condition, stock price, or gas price among others.[28]

As the capabilities of intelligent personal assistants became more diverse and complex, many systems adopted an agent-oriented architecture. For instance, RADAR is a calendar and email management system that can extract task-relevant information from emails and provides guidance to the user on what task to perform (e.g., scheduling a conference room). It is also capable of optimizing the calendar schedule, make decisions autonomously, and negotiate schedules with other users.[22,29] Different tasks are performed by different task-specific specialists, and the coordination is controlled by the central task manager.[30] Similarly, a virtual assistant called CALO also uses an agent-oriented architecture with asynchronous messaging-passing.[23] CALO can also perform various time and task management tasks. Unlike RADAR, however, CALO´s framework utilizes a cognitive model called Belief-Desire-Intention (BDI) model.[31] BDI is a model of human practical reasoning that describes how an agent makes a commitment to certain plans in the presence of long-term goals. Beliefs, desires and intentions represent the informational state, the motivational state, and deliberative state of the agent, respectively. CALO extends the BDI model to support a delegative model, where the user assigns tasks to an agent. When a task is delegated to an agent, it adopts the task as a goal and generates specific plans to achieve that goal. The expressivity of the model allows an agent to process an infeasible set of tasks delegated to it. When there is a misalignment between the agent's and the human user's beliefs, then a refinement process is need to establish mutual belief and derive an appropriate goal. The user can interact with CALO not only through a visual interface but also using natural language.

While larger focus was put on developing intelligent personal assistants that perform everyday tasks such as managing emails and schedules, there also exist intelligent assistants that target more domain-specific tasks. For example, h-Life is a knowledge-based system that is designed as a personal health advisor.[32] It provides personalized health information, gives advices about specific health problems, and proposes plans for improving lifestyle. HealthPal is another personal medical assistant that monitors various health conditions and posts alerts when there are abnormal indications.[33] HealthPal utilizes natural language interaction for communicating with the user.

Intelligent assistants have also been developed and tested for aerospace and defense applications. For example, CAMA is an intelligent assistant for ensuring a pilot's situational awareness during a flight.[25] The agent is capable of understanding the flight situation in light of the goal, understand the intent of the pilot and his or her possible errors. Then based on those observations, it initiates human-like communication with the pilot to ensure their situational awareness. In the case of traffic conflict, it signals warning signs and generates proposals on how to resolve the conflict. A similar type of assistant for multi-UAV guidance have also been developed.[34] It contains environment models to determine the most urgent task and the operator workload. Based on that information, the agent selects appropriate "desires" (goals) and corresponding actions. COGAS is another intelligent assistant, and supports crew of a combat information center in a Navy ship.[35] COGAS combines multiple sources of information to establish relevant goals and selects tasks associated with each goal. Examples of tasks that can be carried out are obtaining and displaying various track data from various sensors and radars, and identifying an unknown object around the Naval ship.

One of the important capabilities of the intelligent assistants discussed so far is providing the necessary or useful information to the user. The agents signal warning signs or generate notifications when there is a conflict in the user's request, when the user's action violate some pre-defined rules, or when the situation requires an action from the user. This greatly reduces the cognitive load of the user, as he or she does not have to remember or check all relevant factors when making a decision. Rather, they can direct their attention to more urgent or important aspects of the goal they are trying to achieve. Then the agent will provide necessary information when the user needs it. We believe the same argument can be applied to complex systems design, where the system designers are often put under a large cognitive load. This research aims to develop an intelligent assistant that can reduce the cognitive load of the system designers by providing information relevant to the current progress of the search.

III. System Overview

III.A. Overall Architecture

This section describes the overall architecture of Daphne. Daphne has a modular structure, with different modules connected through interfaces with the main server module, called the Daphne Brain. Each module communicates with the others using an asynchronous messaging scheme. The messages may contain data, queries, or requests to perform tasks. The basic architecture of Daphne is shown in Figure 1. The dashed line separates the front end from the back end. The front end consists of different interfaces that the user can use to communicate with Daphne. The back end consists of 1) the Daphne Brain, which ensures proper communication between all modules; 2) the Critic Agent, which provides feedback to the user about the architectures they propose; 3) the Analyst Agent, a question-answering System that processes incoming questions, sends requests to other agents, and forms an answer for the user with the responses obtained from the agents; 4) the Data Mining Agent, which can run machine learning algorithms on the dataset for insight generation; 5) the Architecture Evaluation agent, which encapsulates the objective function and can thus evaluate an architecture. The three knowledge bases (expert knowledge, historical database, current dataset) are also part of the back end. The following sections will describe each of these modules in more detail.

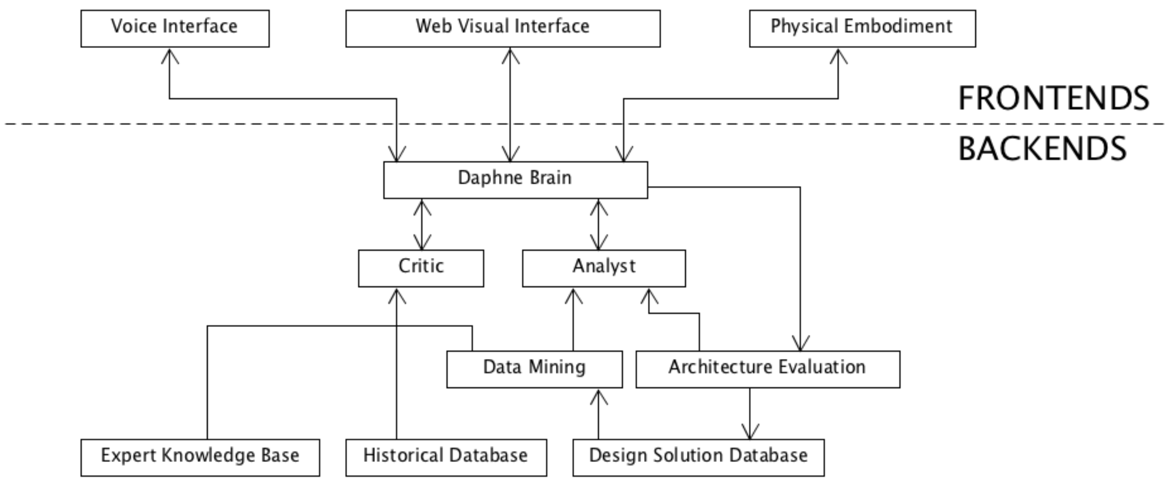

Figure 1. The overall architecture of Daphne

III.B. Sources of Knowledge

Intelligent assistants require a source of knowledge to be able to make inferences, make decisions, and eventually assist humans. These sources can vary from classical databases and text corpus to a knowledge base, as knowledge can be represented in many different forms. Daphne uses data, facts and rules from three different sources: a historical database with information about past Earth Observation Satellite missions, a knowledge base that incorporates rules and facts defined by domain-experts, and a dataset generated by synthesizing and evaluating many alternative architectures.

III.B.1. Historical database

One of the sources of knowledge for Daphne is a database of 567 past and planned Earth observing satellite missions, thus providing a complete picture of the history of Earth observing missions. This database contains information on facts such as mission orbits, launch dates, instruments, and the kinds of measurements taken by those instruments. There is also detailed information about the instruments, including the overall type of technology (e.g., lidar), the accuracy, spatial resolution, spectral region and others.

Most of the information in this database was directly taken from the online CEOS database.[36] The data was obtained by means of a web scraper and crawler that runs over CEOS Database pages and extracts the information of interest. A data processing pipeline was used to curate some of the information and facilitate its usage by the system.

The database was implemented both as a SQL Database, using PostgreSQL, and as an ontology in the RDF format.

III.B.2. Expert Knowledge base

The expert knowledge base contains domain-specific knowledge about how to design Earth observing satellite missions encoded as logical rules that have if-then structure. For example, there is a rule that states that UV/VNIR chemistry spectrometers should be used for missions that fly in afternoon sun-synchronous orbits rather than morning or dawn-dusk orbits. This is because the dawn-dusk orbit has suboptimal illumination conditions for this kind of instrument, and the pollution typically peaks in the afternoon as opposed to the morning.[37] Another example is that an active and a passive instrument that use the same frequency band should not be used in the same spacecraft, as they may interference with each other. While these rules may be considered to be relatively simple, having access to them when they are needed during the architecting process may reduce the cognitive load of a system designer, who has to consider many other factors, and thus lead to improved performance.

III.B.3. Knowledge Extracted from Data Mining

Daphne also draws information from the knowledge extracted by running data mining on the current design space. The design space is populated by a large number of alternative architectures that are defined by their input design variables and the corresponding objectives. The architectures can be generated using various sampling methods such as random sampling and Latin Hypercube sampling, which can sample the feature space in an unbiased way, or optimization algorithms, which are more likely to generate high-quality solutions, thereby introducing bias. Then we apply a data mining algorithm called association rule mining.[38] Association rule mining extracts knowledge in the form of logical rules. The major advantage of logical rules is that they are very similar to how humans reason, and therefore they are easy to understand.[14] Using association rule mining, we extract the driving features. The driving features are defined as a set of features that generally drives solutions to a desired target region in the objective space.[13] If a design has such features, it is likely to also have favorable objective values.

III.C. Roles of Daphne

III.C.1. Analyst/Question Answering System

The Analyst is one of the key differences of our system compared to other decision support tools available to system architects. The goal of the Analyst is to answer diverse questions from the user. It is implemented as a question-answering system. The Analyst is thus a much more intuitive way of interacting with a computer compared to creating code to develop charts, making database queries, or just using a mouse, which are the most common ways to interact with a decision support tool

Sentences are accepted from the user in natural language (English). The questions are classified according to their intent, which represents the objective the user wants the system to accomplish by speaking that sentence. To classify the intent of the natural language queries, a deep learning model based on a Convolutional Neural Network (CNN)[39] was chosen over simpler methods such as regular expressions because of its greater tolerance to input variability and the lower cost of adding new intents to the set. The size of the word filters was set to 3, 4 and 5 words and the number of filters per word size was set to 100. According to Zhang and Wallace,[40] these values give consistently good results on different datasets. Then the model is used to classify the input utterance into different categories at two different points in the pipeline: first to decide the intent of the sentence and later, if the sentence is a question to classify the type of the question. There is also a spelling corrector, which uses the Sellers' algorithm[41] to search for close words, correcting wrongly spelled commands and questions so they make sense to the different services which need to act based on the input.

There is usually a trade-off between accuracy and the number of question answered in a QA system: some systems choose to answer only a small set of questions with great accuracy while others decide to answer as much questions as possible, with the result of lowered accuracy. This can be seen both in open-domain solutions like Aqqu[42] and YodaQA,[43] with the first having a lower recall (number of questions answered) and a high accuracy (number of correctly answered questions) and the second exhibiting the opposite behavior over general datasets such as WebQuestions.[44] Domain specific systems like PRECISE,[45] which is basically a translator from natural language to SQL queries, work over a much smaller set of questions. As a result, it shows much higher recall and accuracy. Because Daphne has a well-defined application domain, we can assume that the types of questions that the user will ask are limited as well. Therefore, we target achieving high accuracy rather than high recall by implementing the Analyst in a similar fashion to PRECISE for generating database queries from natural language questions.

The functional flow of how the Analyst responds to a user question is as follows:

1. The input question or command is obtained as either a typed or spoken sentence. The system assumes that the recognized sentence by the voice Speech To Text (STT) system is correct and will try to find errors later.
2. The CNN model is used to classify questions and commands into different categories based on what module is responsible for the reponse.
   (a) If the input sentence is a command for any of the backends, it is executed and the results of the action are shown to the user, either as a text message appearing on the screen, a spoken sentence, or other kinds of output supported by the frontends, such as a graph, or motion by the physical embodiment.
   (b) If it is a question, the following steps apply:
      i. The question is classified into one of the types of questions supported by the system. Questions of the same type can be answered by the same query. The queries can be either a database query or an agent query. Some of the examples are: "which missions can measure <measurement>","when was mission <mission>launched" or "which missions are currently flying <technology>". Using a CNN model allows for the wording of these questions to be slightly different without losing recall, which is what used to happen with pure regular expressions based systems.
      ii. Once the question is classified, a feature extractor tries to obtain all the information needed for the query either by matching the sentence to lists of known values for the features, while at the same time correcting mistakes using the statistical model provided by Spacy.[46]
      iii. With the features extracted, the query is built according to the configuration for the specific question. Then, the query is executed.
      iv. Finally, the query results are embedded into a template response which is then returned to the frontend that asked for it using voice, text or multimedia content.

III.C.2. Critic Agent

The aim of the Critic agent is to criticize an architecture proposed by the user, i.e., to provide feedback to the user about the strengths and weaknesses of that architecture. In addition to the criticism, it also provides specific suggestions to the user about how to improve a given architecture.

Using the three different sources of information, there are three different types of critiques or recommendations: those based on expert knowledge (rule-driven), those based on past missions (legacy-driven), and those based on knowledge extracted from design solutions (data-driven).

In the case of rule-driven critique, the agent notifies the user if a given design violates a set of rules defined a priori. These rules can be considered as basic design principles or heuristics that domain experts use for coming up with good designs (e.g. two instruments using the same frequency should not be used in a same spacecraft if at least one of them is active).

In the legacy-driven critique, Daphne will notify the user if a mission with a similar configuration (similar types and classes of measurement instruments) has been flown in a similar orbit in the past. This would indicate that since the similar type of mission has been proven successful, and a given design is also likely to perform as well. The motivation for this approach is similar to that of the case-based reasoning,[47] which is one of the popular reasoning methods used in artificial intelligence. Conversely, if a similar mission has never been flown in 50 years of space-based Earth observation, this is also a powerful argument suggesting that perhaps there is a flaw in the design of such mission.

Under the last type of critique (data-driven), Daphne notifies the user if a given design shares the same features that are found frequently among good designs. Again, while sharing a certain design feature with most good designs is no guarantee for success, it can be helpful information. Similarly, Daphne can use this information to suggest changes to a user-defined architecture, based on the good features found so far.

III.D. Modes of Interaction

III.D.1. Voice-based interaction

The voice interaction in Daphne is bidirectional: it can accept voice input and some responses can be spoken out, specially those which are similar to queries and can be answered as a short response to the question.

For the voice recognition we rely on the Web Speech API, which uses Google STT system to work.[48] The current Text To Speech (TTS) service also relies on Google, while also using the ResponsiveVoice.JS library[49] for ease of use.

III.D.2. Physical Embodiment

The design of Daphne's physical embodiment is presented in Figure 2. This robot will be built using 3d printing technology, which means that anyone will be able to build their own version of it. The robot features an LCD screen, which can be used to either show facial expressions or display any type of data (e.g. images and graphs). The robot also has smooth, variable speed, pan and tilt moves. These features will be used to track the user, but also to increase its expressiveness (e.g. nod its head).

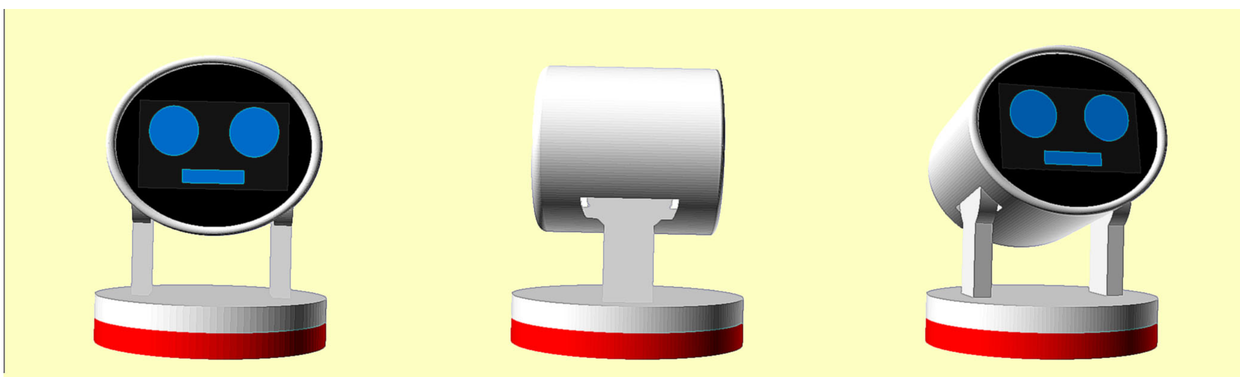

Figure 2. 3D renderings of the physical embodiment

IV. Use Case Scenario: A Constellation for Operational Climate Monitoring

To test the efficacy of Daphne to support the design of an Earth observing system, we used it to work on a real-world problem of architecting an Earth observing satellite system to perform operational monitoring of the Earth's climate, as described in.[26] Specifically, the goal is to design a constellation of satellites that maximizes the satisfaction of a set of 371 climate-related measurement requirements while minimizing the life-cycle cost. These measurement objectives are taken from the World Meteorological Organization's OSCAR (Observing Systems Capability Analysis and Review Tool) database [a]. The lifecycle cost model is taken from Apgar et al.[50]

The design problem is formulated as an assignment problem between instruments and orbits. It is assumed that all instruments assigned to an orbit are flown in the same spacecraft. In other words, we want to find the optimal assignment of different measurement instruments into spacecraft that will fly in different orbits. In this problem, there are 12 candidate instruments and 5 candidate orbits. Because every instrument can be assigned to each orbit or not, there are a total of $2^{12\times 5} = 2^{60}$ possible architectures.

For the full version of this paper, we will report a detailed use case scenario of how Daphne can be applied to help the system designer solve the above mentioned system architecting problem. The use case scenario

[a] http://www.wmo-sat.info/oscar/

will describe under what conditions the user may make a request to Daphne for information, what type of information can be provided, how they are presented to the user, and how this will eventually benefit the user by reducing the cognitive load.

Acknowledgments

This project is partially supported by National Science Foundation Grant Number CMMI-1635253.

References

[1]Maier, M. W. and Rechtin, E., *The Art of Systems Architecting*, CRC press, New York, 2000.

[2]Woodruff, M. J., Reed, P. M., and Simpson, T. W., "Many objective visual analytics: rethinking the design of complex engineered systems," *Structural and Multidisciplinary Optimization*, Vol. 48, No. 1, 2013, pp. 201–219.

[3]Ko, A., Daniel, J., Keel, W., and Baines, A., "Whitespace Exploration: The Next Step in Searching the Design Space," *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2017, p. 0971.

[4]Stump, G. M., Yukish, M., Simpson, T. W., and Harris, E. N., "Design space visualization and its application to a design by shopping paradigm," *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2003, pp. 795–804.

[5]Eddy, J. and Lewis, K. E., "Visualization of multidimensional design and optimization data using cloud visualization," *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2002, pp. 899–908.

[6]Chiu, P.-W. and Bloebaum, C. L., "Hyper-Radial Visualization (HRV) method with range-based preferences for multi-objective decision making," *Structural and Multidisciplinary Optimization*, Vol. 40, No. 1-6, 2010, pp. 97.

[7]Knerr, N. and Selva, D., "Cityplot: Visualization of High-Dimensional Design Spaces With Multiple Criteria," *Journal of Mechanical Design*, Vol. 138, No. 9, 2016, pp. 091403.

[8]Chiu, P.-W. and Bloebaum, C., "Visual Steering for Design Generation in Multi-objective Optimization Problems," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, 2009, p. 1167.

[9]Stump, G., Lego, S., Yukish, M., Simpson, T. W., and Donndelinger, J. A., "Visual steering commands for trade space exploration: User-guided sampling with example," *Journal of Computing and Information Science in Engineering*, Vol. 9, No. 4, 2009, pp. 044501.

[10]Yan, X., Qiao, M., Simpson, T., Li, J., and Zhang, X., "Work-Centered Visual Analytics to Support Multidisciplinary Design Analysis and Optimization," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 5662.

[11]Watanabe, S., Chiba, Y., and Kanazaki, M., "A proposal on analysis support system based on association rule analysis for non-dominated solutions," *Evolutionary Computation (CEC), 2014 IEEE Congress on*, IEEE, 2014, pp. 880–887.

[12]Cervone, G., Franzese, P., and Keesee, A. P., "Algorithm quasi-optimal (AQ) learning," *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2, No. 2, 2010, pp. 218–236.

[13]Bang, H. and Selva, D., "iFEED: Interactive Feature Extraction for Engineering Design," *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2016.

[14]Newell, A., Simon, H. A., et al., *Human problem solving*, Vol. 104, Prentice-Hall Englewood Cliffs, NJ, 1972.

[15]Selva, D., Abello, C., and Hitomi, N., "Preliminary experiments with learning agents in an interactive multi-agent systems architecture tradespace exploration tool," *Systems Conference (SysCon), 2015 9th Annual IEEE International*, IEEE, 2015, pp. 445–452.

[16]Michalski, R. S., "Learnable evolution model: Evolutionary processes guided by machine learning," *Machine learning*, Vol. 38, No. 1, 2000, pp. 9–40.

[17]Hitomi, N., Bang, H., and Selva, D., "Extracting and Applying Knowledge with Adaptive Knowledge-Driven Optimization to Architect an Earth Observing Satellite System," *AIAA Information Systems-AIAA Infotech@ Aerospace*, 2017, p. 0794.

[18]Gaur, A. and Deb, K., "Adaptive Use of Innovization Principles for a Faster Convergence of Evolutionary Multi-Objective Optimization Algorithms," *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, ACM, 2016, pp. 75–76.

[19]Allen, J., Guinn, C. I., and Horvtz, E., "Mixed-initiative interaction," *IEEE Intelligent Systems and their Applications*, Vol. 14, No. 5, 1999, pp. 14–23.

[20]Fong, T., Thorpe, C., and Baur, C., "Collaboration, dialogue, human-robot interaction," *Robotics Research*, Springer, 2003, pp. 255–266.

[21]Segal, R. B. and Kephart, J. O., "MailCat: An intelligent assistant for organizing e-mail," *Proceedings of the third annual conference on Autonomous Agents*, ACM, 1999, pp. 276–282.

[22]Freed, M., Carbonell, J. G., Gordon, G. J., Hayes, J., Myers, B. A., Siewiorek, D. P., Smith, S. F., Steinfeld, A., and Tomasic, A., "RADAR: A Personal Assistant that Learns to Reduce Email Overload." *AAAI*, 2008, pp. 1287–1293.

[23]Myers, K., Berry, P., Blythe, J., Conley, K., Gervasio, M., McGuinness, D. L., Morley, D., Pfeffer, A., Pollack, M., and Tambe, M., "An intelligent personal assistant for task and time management," *AI Magazine*, Vol. 28, No. 2, 2007, pp. 47.

[24]Chen, K.-J. and Barthès, J.-P., "MemoPA: Intelligent Personal Assistant Agents with a Case Memory Mechanism," *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, 2007, pp. 1357–1367.

[25]Onken, R. and Walsdorf, A., "Assistant systems for aircraft guidance: cognitive man-machine cooperation," *Aerospace Science and Technology*, Vol. 5, No. 8, 2001, pp. 511–520.

[26]Selva, D., "Knowledge-intensive global optimization of Earth observing system architectures: a climate-centric case study," *SPIE Remote Sensing*, Vol. 9241, 2014, pp. 92411S–92411S.

[27]Engelbart, D. C., "Augmenting human intellect: a conceptual framework (1962)," *PACKER, Randall and JORDAN, Ken. Multimedia. From Wagner to Virtual Reality. New York: WW Norton & Company*, 2001, pp. 64–90.

[28]Kongthon, A., Sangkeettrakarn, C., Kongyoung, S., and Haruechaiyasak, C., "Implementing an online help desk system based on conversational agent," *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, ACM, 2009, p. 69.

[29]Modi, P. J., Veloso, M., Smith, S. F., and Oh, J., "Cmradar: A personal assistant agent for calendar management," *Agent-Oriented Information Systems II*, Springer, 2005, pp. 169–181.

[30]Garlan, D. and Schmerl, B., "The RADAR architecture for personal cognitive assistance," *International Journal of Software Engineering and Knowledge Engineering*, Vol. 17, No. 02, 2007, pp. 171–190.

[31]Rao, A. S., Georgeff, M. P., et al., "BDI agents: From theory to practice." *ICMAS*, Vol. 95, 1995, pp. 312–319.

[32]Nanou, D., Poulis, A., Likas, A., and Fotiadis, D., "h-Life: an intelligent personal health assistant," *Health Informatics Journal*, Vol. 8, No. 1, 2002, pp. 20–28.

[33]Komninos, A. and Stamou, S., "HealthPal: an intelligent personal medical assistant for supporting the self-monitoring of healthcare in the ageing society," *Proceedings of the UbiHealth*, 2006.

[34]Donath, D., Rauschert, A., and Schulte, A., "Cognitive assistant system concept for multi-UAV guidance using human operator behaviour models," *HUMOUS10*, 2010.

[35]Özyurt, E. and Döring, B., "A cognitive assistant for supporting air target identification on navy ships," *Cognitive Technical Systems: Modeling and Simulation*, 2012.

[36]CEOS, "THE CEOS DATABASE: MISSION, INSTRUMENTS AND MEASUREMENTS," 2017.

[37]Selva, D., "Knowledge-intensive global optimization of Earth observing system architectures: a climate-centric case study," *SPIE Remote Sensing*, International Society for Optics and Photonics, 2014, pp. 92411S–92411S.

[38]Agrawal, R., Imieliński, T., and Swami, A., "Mining association rules between sets of items in large databases," *Acm sigmod record*, Vol. 22, ACM, 1993, pp. 207–216.

[39]Kim, Y., "Convolutional Neural Networks for Sentence Classification," *CoRR*, Vol. abs/1408.5882, 2014.

[40]Zhang, Y. and Wallace, B. C., "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," *CoRR*, Vol. abs/1510.03820, 2015.

[41]Sellers, P. H., "The theory and computation of evolutionary distances: Pattern recognition," *Journal of Algorithms*, Vol. 1, No. 4, 1980, pp. 359 – 373.

[42]Bast, H. and Haussmann, E., "More accurate question answering on freebase," *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ACM, 2015, pp. 1431–1440.

[43]Baudiš, P., "YodaQA: a modular question answering system pipeline," *POSTER 2015-19th International Student Conference on Electrical Engineering*, 2015, pp. 1156–1165.

[44]Berant, J., Chou, A., Frostig, R., and Liang, P., "Semantic Parsing on Freebase from Question-Answer Pairs." *EMNLP*, Vol. 2, 2013, p. 6.

[45]Popescu, A.-M., Etzioni, O., and Kautz, H., "High precision natural language interfaces to databases: a graph theoretic approach," *Proceedings of the 2003 Conference on Intelligent User Interfaces*, 2003.

[46]ExplosionAI, "Entity recognition — Spacy," 2017.

[47]Xu, L. D., "Case based reasoning," *IEEE potentials*, Vol. 13, No. 5, 1994, pp. 10–13.

[48]W3C, "Web Speech API Specification," 2012.

[49]Learnbrite, "ResponsiveVoice for Text to Speech," 2017.

[50]Apgar, H., Bearden, D., and Wong, R., "Cost Modeling," *Sp. Mission Anal. Des.*, edited by W. J. Larson and J. R. Wertz, chap. 20, Microcosm, Hawthorne, CA, 3rd ed., 1999, pp. 783–820.