Insights for Configuration in Natural Language

Andres Felipe Barco Santa, Élise Vareilles, César I. Osorio

Insights for Configuration in Natural Language

Andrés F. Barco[1] and **Élise Vareilles**[2] and **César I. Osorio**[1]

Abstract. Usually, in configuration processes, customers interact with a decision support system, also named configurator, by explicitly selecting components or required functionalities through a written series of questions, until the complete configuration is done and the desired product is defined. The interactions during a configuration process may vary vastly depending on the customers' knowledge about the product and his/her understanding of its potential functionalities. However, configurators are not conceived for making a difference between expert and inexpert customers as interfaces and input information are all expected to be the same for everyone. This paper discusses how *natural language* can enhance configuration process by making possible for customers to express their desires, needs and preferences in natural language, and for configurators to interpret their words and better help them to find relevant solutions. This kind of configuration process could have as foundations an expert systems that that maps speech into constraints and objectives. We present the artificial intelligence trends motivating our research, an initial architectural design and potential applications of the research.

1 Introduction

For several decades now, customers want to bring a personal touch to their products to make them special and unique. To meet this growing demand of personalization, companies nowadays no longer offer standard products, but more and more personalizable ones [1]. Thanks to the Web technologies and dedicated decision support systems, named configurators, this personalization is done directly and interactively online [1]. Customers can play with the wide range of choices and options offered by companies: they can assemble, cut, color, choose, ..., and visualize the result of their desires and ultimately order it, all in few clicks and minutes.

This concept of personalization or configuration of products consists in assembling modules or predefined components, to produce a unique and specific product [10]. For businesses, this is a way to offer personalized products to stand out from the competition and build customers' loyalty through more accurately reflecting their tastes and needs.

Interactions between potential customers and configurators, become now one of the key aspects of configuration problems [22]. Nevertheless, often the configuration relies in a long data capturing process. Normally, to configure the product object of desires, any potential customer has to:

1. Face the increasing range of choices and options without being completely able to focus on his/her essential items,
2. answer a predefined series of questions, always in the same order whatever his/her knowledge and needs about the product,
3. express his/her needs and preferences in such a way they fit the predefined set of choices and options proposed online by configurators, and
4. click to select the relevant functions or components meeting the best of his/her needs.

All these facts gathered make the configuration of products more and more a counter-intuitive process. Also, current interaction makes no difference between expert and non-expert user as most of the input mechanism (such of as graphical windows, drawings, text fields) are all expected to be the same for everyone [2].

This paper discusses how the mature techniques from AI may be used to allow a more natural interaction between customers and configuration systems. In essence, we describe the future of configuration in which natural language interactions could replace the traditional one based on writing, explicit selection of items and rigid series of questions. More precisely, we argue that current AI advances like natural language processing could help customers to express their needs implicitly in *speech* format (e.g. "I need a cheap laptop with good video card") while an expert system could infer the requirements and set the goals of the configuration (e.g. video card $\geq$ good). We call this kind of configuration *Configuration in Natural Language*. We also present an idea of architecture of such an expert system.

The paper is divided as follows. In Section 2, the traditional and future customer-system interactions in configuration are discussed. In section 3, a software architecture based on current advances in AI is introduced. In Section 4, some of the mathematical frameworks from which the configuration in natural language can take advantage are discussed. Finally, in Section 5, some applications of the research and conclusions are presented.

2 Configuration Interactions: Past and Future

We present in this section the interaction typically made in configuration systems and contrast it with the idea of configuration in natural language proposed in this paper.

2.1 Traditional Configuration Interactions

In most of the cases, configuration systems follow a series of iterative steps that guide the customer and help him/her to progressively configure his/her own product. These steps, that we call the *standard way of configuration*, are as follows:

1. The system shows some sets of components and functionalities to the customer in a predefined way,

[1] Universidad de San Buenaventura Cali. Santiago de Cali, Colombia. email: {anfelbar,ciosoriod}@usbcali.edu.co
[2] Université de Toulouse, Mines Albi. Albi, France. email: elise.vareilles@mines-albi.fr

2. The customer either selects the most relevant component or functionality which meets his/her needs and desires, or specifies a value for the most important criteria (such as the price he/she is willing to pay for the product),
3. The system removes or assigns the set of remaining components and functionalities according to the set of constraints in the product,
4. The system computes the price of each possible product and valuates their criteria.

In step 2, the selection is usually done manually by clicking on the relevant item via a mouse-click or a touch screen, or by inputting its specific value directly from a keyboard. At the end of the standard way of configuration, the customer selects his/her unique product and orders it.

2.2 Future Configuration Interactions

The configuration in natural language changes the interactions between customers and configurators. Customers will be able, whatever their requirements and knowledge about the product, to express better in a more natural way their preferences, desires and needs and configure faster their own products. The significant difference between traditional and forthcoming configuration interactions lies in the way of expressing and capturing customers' needs and goals. The steps for the configuration in natural language are as follows:

1. The system welcomes the customer.
2. The customer writes or says what he/she wants or needs by using his/her own words.
3. The system infers the set of components and functionality the user wants or needs, and the goal of the configuration.
4. The system removes or assigns the set of remaining components and functionalities according to the set of constraints in the product.
5. The system computes the price of each possible product and valuates their criteria.

To exemplify this kind of configuration, limit us to written interactions in natural language via a keyboard or similar device. Three examples of such interactions when configuring a computer, and the respective responses of the inferring engine, are:

- **Customer express:** "I want a really fast computer but not too expensive"
  System infers: *Component*(processor, speed, high) *Goal*(computer, cost, low)
- **Customer express:** "I just want to play video games, preferably not heavy so I can carry it easily"
  System infers: *Component*(video_card, processing, high) *Goal*(computer, weight, low)
- **Customer express:** "I need to write my texts"
  System infers: *Component*(keyboard, comfort, high) *Goal*(none, none, none)

As expected, inferring components or functionalities and configuration objectives is a major challenge. On the first hand, the universe of words used by humans to express the same thing may be vast. Second, the way to build expressions may vary largely depending on academic background, experience, state of mind and mood, to name a few. Finally, it is difficult to set a difference between components and functionalities, and goals; critical to reach an appropriate configuration solution. For instance, in the first of the two previous examples, both the processor speed and the computer cost may be seen as configuration goals. For tackling this challenge, we propose an expert system architecture built upon AI trends.

3 AI Trends-based Architecture

The field of AI is generating broad interest. Technological advances using AI techniques, such as the DeepMind GO system developed by `Google`™ [19] and the `IBM Watson`™ analytic system [11], draw the attention of the academic and non-academic world on the innovative role of mathematical models from computer science and philosophy. Current trends show that the use of AI and other related fields are being widely used sectors such as economy, health, transport industry, aviation and games, among others [20].

The configuration in natural language is motivated by the recent advances in AI and by industrial trends, in particular the growing interest in the construction of machines that understand human emotions and act according to the interaction with human [16]. It is sought that the machines assist decision-making processes whereas the understanding of human emotions helps to improve the expert systems behavior and interaction [8]. This idea, known as *Human Aware AI*, is not new in configuration as it has been used as a goal in different configuration systems (see for instance [3]). Further, the idea of understating or inferring user needs has been widely study in the plan recognition problem [6]. Nonetheless, to the best of our knowledge, current advances in natural language use remain to be adopted in configurators implementations.

Within the human aware AI field, the Natural Language Understating (NLU) [4] and Natural Language Processing (NLP) [12] draw attention for its capabilities of human computer interaction. In essence, NLU and NLP systems allow the user to ask questions in everyday language and try to *understand* these questions in order to return appropriated answers. Typically, these systems makes some hypothesis according to the question and a knowledge base, such as Internet, and then process an output. This is akin to the problem of plan recognition, i.e., knowing the user's plans and goals [6]. Further, if these systems are improved with natural language generation (NLG) in order to produce responses, the system then becomes a question answering system (QAS) [13]. These systems were conceived to received provide argued answers to user queries. From these systems, `WolframAlpha`™ [7] and `IBM Watson`™ [11] present the more innovative results for configuration as these systems are able to recognize some information in form of requirements within informal speech in text format.

To illustrate these capabilities, Figure 1 shows the result when querying *"I want a computer of less than 1000 dollars."* in the `WolframAlpha`™ system. To construct a response, the system maps the input into a more elaborated query, encoding the query thus making a syntactic and semantic analysis. Nevertheless, as the system does not focus on inferring mathematical notions, it is easily confused by adding words that add relevant constraints. For instance, no result shown when querying the same computer but `Dell`™ manufactured; *"I want a dell computer of less than 1000 dollars"*. We consider that this is a major drawback in the system when addressing configuration problems.

The `IBM Watson`™ system works similarly to the `WolframAlpha`™. It encodes a query by applying automated reasoning, machine learning and several other techniques to analyze the speech. One of the more innovative applications of the `IBM`

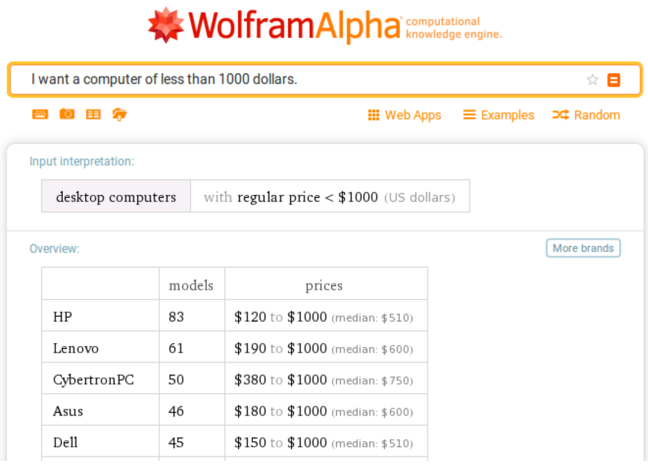

Figure 1. Output example of `WolframAlpha`™ system.

`Watson`™ system is the personality characterization from written speech. To do this, in addition to question answering, different techniques of sentiment analysis [21] are applied. This kind of analysis may be useful to infer expertise level of the user and then behave accordingly.

In spite of the recent advances of NLU, NLP and QAS, these are not well-suited for addressing configuration problems given that they do not focus on constructing the mathematical notions (constraints and objectives) needed in most configuration problems. Besides, question answering systems are too powerful in the sense they are of general purpose having different question types and extensive knowledge bases. At the other end of the spectrum, specific applications using question answering systems technology do not necessarily deal with extensive vocabularies, hypothesis and so on, as these applications are domain-dependent. Ergo, its underlying mechanism may be simpler although more robust and may count with reduced knowledge, question types and small knowledge bases. In consequence, we have devise an architecture, presented in Figure 2, for implementing configurators that exploits the aforementioned natural language elements. The mandatory module is that of NLU whereas NLP and NLG are optional (used if formatted answers are desired). External services may be attached in order to fulfill specific tasks.

The differentiating element in the architecture is the inference engine. This key element is in charge of discriminating among the set of words those referring to components of the product, functionalities and/or configuration objectives. This is in fact a challenge as different conclusions may be reached from a given sentence.

4 Mathematical Frameworks for Configuration in Natural Language

Unlike NLP and QAS, configuration processes are mapped, generally, into mathematical (optimization) models that unify the customers requirements and problem domain limitations into a single framework. To infer constraints and objectives from informal speech, it is needed an underlying mathematical framework in which such notions are built. In other words, to construct a mathematical formula it is necessary the set of values and operands allowed in the formula. As an example, if constructing inequalities, the expert system knows

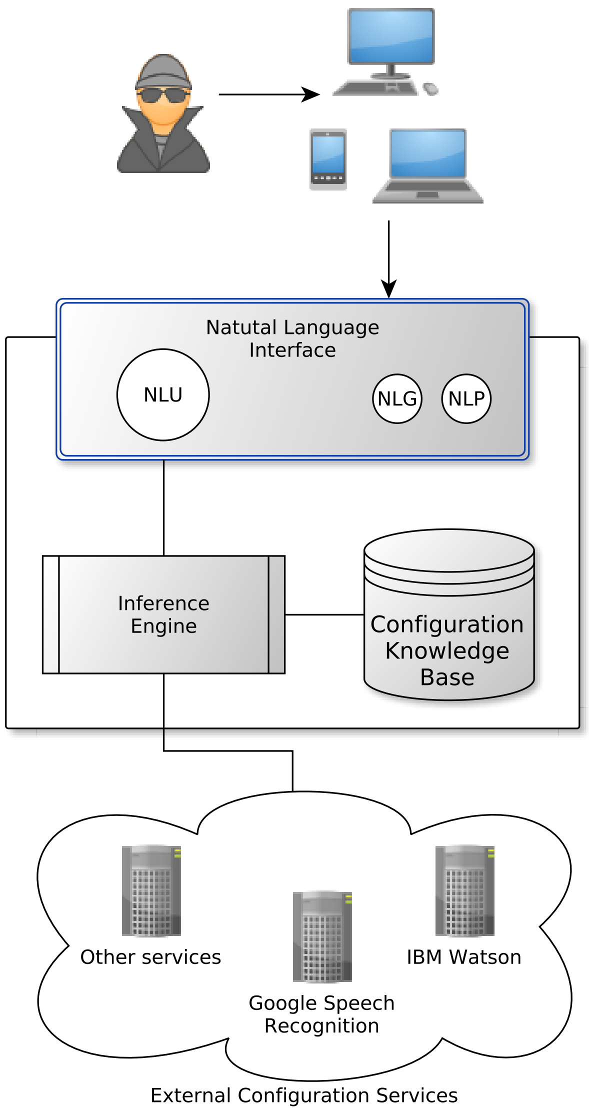

Figure 2. Architectural view of expert system for configuration NL.

that sum, subtract, division and multiplication are allowed as well as inequality symbols. This helps to make the mapping unambiguous. Given that our main innovative application is the configuration in natural language, mathematical frameworks used to tackle configuration problems comes naturally. Here, we briefly describe three of these models.

- The first framework, Constraint Programming (CP), has been identified as a key paradigm in the expansion of applied computer science [17]. CP is part and good representative, of the declarative programming frameworks. This is one of the most used framework to address configuration problems as it suits their constrained nature [15]. First, the knowledge (constraints) that restricts possible configuration of elements (variables) is easily modeled under the declarative framework of constraint satisfaction problems. And second, constraint-based configurators are able to present different solutions to users, often optimal, even when they do not provide all configuration parameters.

- Among the same lines we have integer programming (IP). It is a mathematical optimization in which some or all of the variables are restricted to be integers [18]. Likewise the constraint satisfaction model, it is built from constraints and limitations over variables, although variables do not have a given domain, and objective functions such as minimization and maximization. This model or extensions of it (Mixed IP or Linear IP) have been widely used to reason and solve configuration problems (see for instance [9]).
- Finally, another framework that can be used in configuration in natural language is SAT; determining if a given propositional formula is satisfiable by an interpretation [5]. The inclusion or exclusion of a given feature in a product may be seen as boolean assignments. Thus, it would be possible to deduce if a solution exists by constructing a boolean formula using the requirements of the customers and the configuration knowledge (e.g. compatibility among components). The modeling of a configuration problem under SAT is not new (see for instance [14]).

5 Concluding Remarks

In this paper, we have presented insights on how to enhance the interaction between customers and configuration systems into something called configuration in natural language. In summary, we have discussed the possibility of using AI techniques, namely, human-aware AI to implement an expert system that infers constraints and objectives from a speech input.

We have built out the idea around the concepts of NLU, NLP and QAS. Then, we argue that although current systems have reached a stable development, there is still research to be done when inferring mathematical notions from informal sentences, i.e., everyday language. In addition, we have argued that advanced question answering systems such as `IBM Watson`™ are not yet well suited for configuration problems as they are not intended to map implicit requirements into mathematical models.

To give a view of the mathematical models that can prove useful to construct the proposed expert system, we have briefly described three techniques from AI and operational research. These, techniques, namely, constraints programming, integer programming and boolean satisfiability, have been used to solve different configuration and optimization problems and are in our point of view interesting models to construct a configurator in natural language.

The main application of inferring constraints and goals from speech is linked to the mathematical modeling of real-world problems. Simply stated, the inference system may be used to build mathematical models to solve linear problems, discrete optimization problems and probabilistic problems, among others. Further, specific properties from each mathematical framework may be exploited, such as the expressivity of declarative approaches like logic and constraint programming. Intuitively, many more applications exists; those in which user requirements, preferences, limitations or objectives are needed. For instance package managers in unix-based operating systems. Typically, a package manager from a Unix-based system must be asked to search or install a specific package. Using a expert systems that maps speech into constraints and objectives, a manager would accept inputs like *"I need a powerful UML diagram editor"* to present some potential editors to be installed. Further, arguments used by such managers can be replaced by everyday words thus preventing the non-expert user to learn the specifics of the programs.

REFERENCES

[1] L. Ardissono, A. Felfernig, G. Friedrich, A. Goy, D. Jannach, M. Meyer, G. Petrone, R. Schafer, W. Schutz, and M. Zanker, *Personalising on-line configuration of products and services*, 225–229, Thiel, S., Pohl, K., Lyon France, 2002.

[2] Liliana Ardissono, Anna Goy, Matt Holland, Giovanna Petrone, and Ralph Schäfer, *Customising the Interaction with Configuration Systems*, 283–287, Springer Berlin Heidelberg, 2003.

[3] Virginia E. Barker, Dennis E. O'Connor, Judith Bachant, and Elliot Soloway, 'Expert systems for configuration at digital: Xcon and beyond', *Commun. ACM*, **32**(3), 298–318, (March 1989).

[4] M Bates, 'Models of natural language understanding', *Proceedings of the National Academy of Sciences*, **92**(22), 9977–9982, (1995).

[5] A. Biere, A. Biere, M. Heule, H. van Maaren, and T. Walsh, *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*, IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009.

[6] Sandra Carberry, 'Techniques for plan recognition', *User Modeling and User-Adapted Interaction*, **11**(1), 31–48, (Mar 2001).

[7] John B. Cassel, *Wolfram—Alpha: A Computational Knowledge "Search" Engine*, 267–299, Springer New York, New York, NY, 2016.

[8] Celso M. de Melo, Stacy Marsella, and Jonathan Gratch, '"do as i say, not as i do": Challenges in delegating decisions to automated agents', in *Proceedings of the 2016 International Conference on Autonomous Agents #38 Multiagent Systems*, AAMAS '16, pp. 949–956, Richland, SC, (2016). International Foundation for Autonomous Agents and Multiagent Systems.

[9] Ingo Feinerer, 'Efficient large-scale configuration via integer linear programming', *Artif. Intell. Eng. Des. Anal. Manuf.*, **27**(1), 37–49, (January 2013).

[10] Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 edn., 2014.

[11] D. A. Ferrucci, 'Introduction to "This is Watson"', *IBM Journal of Research and Development*, **56**(3.4), 1:1–1:15, (May 2012).

[12] Chowdhury Gobinda G., 'Natural language processing', *Annual Review of Information Science and Technology*, **37**(1), 51–89.

[13] L. Hirschman and R. Gaizauskas, 'Natural language question answering: The view from here', *Nat. Lang. Eng.*, **7**(4), 275–300, (December 2001).

[14] M. Janota, *Do SAT Solvers Make Good Configurators?*, 191–195, Thiel, S., Pohl, K., Limerick, Ireland, 2008.

[15] Ulrich Junker, *Configuration.*, Chapter 24 of Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., New York, NY, USA, 2006.

[16] Stacy Marsella and Jonathan Gratch, 'Computationally modeling human emotion', *Commun. ACM*, **57**(12), 56–67, (November 2014).

[17] Francesca Rossi, Peter van Beek, and Toby Walsh, *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*, Elsevier Science Inc., New York, NY, USA, 2006.

[18] G. Sierksma, *Linear and Integer Programming: Theory and Practice, Second Edition*, Advances in Applied Mathematics, Taylor & Francis, 2001.

[19] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis, 'Mastering the game of go with deep neural networks and tree search', *Nature*, **529**, 484–503, (2016).

[20] Peter Stone, Rodney Brooks, Erik Brynjolfsson, Ryan Calo, Oren Etzioni, Greg Hager, Julia Hirschberg, Shivaram Kalyanakrishnan, Ece Kamar, Sarit Kraus, Kevin Leyton-Brown, David Parkes, William Press, AnnaLee Saxenian, Julie Shah, Milind Tambe, and Astro Teller, 'Artificial intelligence and life in 2030', Technical report, Stanford University, Stanford, CA, (September 2016). One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel, Stanford University.

[21] Maite Taboada, 'Sentiment analysis: An overview from linguistics', *Annual Review of Linguistics*, **2**(1), 325–347, (2016).

[22] Mitchell M. Tseng and S. Jack Hu, *Mass Customization*, 836–843, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.