Explainability in Context: Lessons from an Intelligent System in the IT Services Domain

Christine T. Wolf and Jeanette L. Blomberg
IBM Research, Almaden
San Jose, CA, USA
{ctwolf, blomberg} @us.ibm.com

ABSTRACT

We report from an ongoing study of the design, development, and deployment of an intelligent workplace system in the IT services domain. We describe the system, which is designed to augment the complex design work of highly-skilled IT architects with the use of natural language processing (NLP) and optimization modelling. We outline results from our study, which analyzes feedback from architects as they interacted with various prototypes of the system. This feedback focuses on their sensemaking and uncertainty around: *system actions*; *interactivity and system outputs*; and *integration with existing processes*. These findings point to "explanation" as a multi-dimensional requirement. Such multi-dimensionality requires more careful articulation of the different types of explanations needed to support workers as they make sense of and successfully integrate smart systems in their everyday work practice.

CCS CONCEPTS

• Human-centered computing → Computing methodologies; Artificial intelligence

KEYWORDS

Intelligent Workplace Systems; everyday work practices; natural language processing; IT services; requirements analysis

ACM Reference format:

Christine T. Wolf and Jeanette L. Blomberg. 2019. Explainability in Context: Lessons from an Intelligent System in the IT Services Domain. In Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019. 6 pages.

1 Introduction

Recent years have seen a rapid growth in attention to the explainability or interpretability of machine learning (ML) systems, with such issues capturing the imagination of many popular press outlets such as the New York Times [2], Wall Street Journal [3], and Financial Times [6], to name a few. The ability to reason about and understand ML systems is a pressing need that has motivated the creation of research programs (i.e., DARPA XAI) and the development of a number of novel XAI approaches in recent years.

An important gap exists between purely technical approaches to XAI and the particular, situated explainability requirements that arise in real world deployments. What makes a smart system "explainable" for a given context? What are the key enablers (and inhibitors) to end-users' contextual understanding of such systems? How do technical explanations compare to the types of explanations users need to take action? Scholarly attention is needed to chart the interactional aspects of ML interpretability and how sensemaking and coherence emerge dynamically through interactions between users, smart systems, and the context of their deployments. We investigate this topic in the work of IT architects in the IT services domain and report on a field study of the design and development of an intelligent tool to aid architects' solutioning work. The intelligent tool features a number of ML capabilities, including natural language processing (NLP), intended to support the IT requirements definition stage of solution design, and optimization modelling to match technical requirements with IT service offerings.

In this workshop paper, we discuss issues raised by IT solution designers as they interacted with a number of the tool's prototypes. We organize these into three categories: *system actions*; *interactivity and system outputs*; and *integration with existing processes*. In each, different types of explanations are sought by users. The first focuses on making sense of and assessing the accuracy of system actions. Interactivity and system outputs focuses on users' abilities to identify and understand the impact of their actions on system outputs. Integration with existing processes focuses on users' understanding of the system's outputs in relation to established organizational processes. These issues raise implications for the explainability of smart systems, in particular suggesting the need to more clearly articulate the range of explanations needed to make sense of and successfully work with smart systems in everyday work practice.

The remainder of the paper is laid out as follows. First we discuss the industrial context and setting of our study (the IT services domain), our field study, and our field work to date. Then we outline each of the three initial findings from this work. Then we conclude with a summary and our contributions to the ExSS Workshop at IUI'19.

2 Industrial Context and Setting

Our project's broader industrial context is the design of information technology (IT) infrastructure architectures within an overarching IT services procurement process. The typical use case is a medium to large-size organization – a corporation for example, or perhaps a municipal government or a higher-education institution – decides they want to outsource part or all of their IT infrastructure to an external vendor. These services-seeking organizations are the "clients." The client's IT needs are written up into formal requirements within Request for Proposal (RFP) documentation. The RFP is often a mix of digital files – typically including word processing documents, PowerPoint decks, spreadsheets, and schematic diagrams. It is not unusual that complex RFP packets include upwards of 30 or 40 different files, typically including hundreds of pages of content. Less complex deals might include a handful of documents and around a hundred pages of content. Sometimes the RFP process is managed by third party vendors, who might be the ones who prepare the RFP packet and then also handle the bidding process from vendors vying for the IT contact; in other deals, the client manages this process themselves. For some deals, there is a Q&A period, where various vendors may bring clarification questions to the client or the client's representatives. This does not occur in all deals, however, and thus the RFP documentation is a key source of insight into the client's needs and overall goals for the IT services contract.

Our focus is on the work IT service vendors do to prepare architectures and create bids for RFPs which include "solutions" designed by the IT architects. Solution design is today a largely manual process where highly-skilled IT architects read the RFP documentation line-by-line, deconstructing the content to pull out the client's individual technical requirements. Most of the data within the RFP is unstructured text, which architects must transform into more structured formats (e.g., copying a text string from a document into a spreadsheet cell). This structural transformation enables various kinds of downstream analysis, as the text requirements are mapped to their numerical baseline values (e.g., how many units of a given item, as well as if/how those quantities are expected to change over the life of the contract). These requirements and baselines are then further mapped to higher-level IT services frameworks, typically divided among two to three hierarchal classification layers, and then ultimately matched to different sets of IT offerings (bundles of services that meet various service requirements). Offerings coverage is optimized to create a solution bid that maximizes coverage for the client, while taking into consideration other factors and constraints (e.g., cost, pricing, market standards, etc.).

2.1 Field Study: Developing an Intelligent Tool

The context of our field study is an ongoing project that is developing an intelligent tool to aid in the IT solution-building process at a large, global technology services firm, EnterprizeIT[1] [5]. Development of the intelligent workplace system (referred to in this paper as either "ITDezigner" or simply "the system" or "the tool"), follows the Agile software management method, a "continuous delivery" model where initial features of a software application are released, and then iteratively enhanced over time; future feature functions are re-planned and implemented based on cycles of feedback, reflection, and planning. Central to Agile is the active and ongoing involvement of stakeholders throughout the development process, of whom a key constituency are the intended users of the application.

In this paper, we report on insights gathered from feedback provided by ITDezigner's user community, IT architects within EnterprizeIT and those who manage the solution design work, which they provided after interacting with various prototypes of the system. As ITDezigner supports a complex process, it is a complex tool featuring several different functionalities meant to support the solutioning design process (including requirements extraction and classification, the optimization of requirements to the firm's offerings, and the automatic creation of collateral material describing the suggested solutions).

We report here on two user feedback programs that ran from October 2017 to July 2018. The first program (Phase I)

[1] All proper names in the paper are pseudonyms.

gathered exploratory feedback on early prototypes of the system and included a small group of users. The second author conducted semi-structured interviews with the Phase I group of users who had interacted with the early prototype of the system between October and December 2017. In total, Phase I included interviews with eight (8) architects and solution managers, averaging one hour in length.

The second program (Phase II) was an "early adopter" program, designed to emphasize and evaluate different components of the tool at different phases. Feedback was solicited from a larger cohort of users and combined both usability testing (looking for system bugs/defects and whether the system was working as designed) and usefulness (evaluating how well the system aligned with the work practices of the solutioning team, evaluating whether the system was fit for purpose). Both the first and second authors carried out Phase II, which ran from January to July 2018 and involved sessions with seventeen (17) individuals in the early adopter program. These sessions were 1 hour long, and included a semi-structured interview portion (discussing the current work practices) and then involved real-time use of the tool, where users were asked to share their screen with the researchers and complete a series of tasks within the system while using the "think aloud" method [5]. In addition to data gathered from these individual sessions, the authors also held focus group sessions with members of the broader early adopter cohort (three (3) focus groups with 8, 10, and 10 architects attending each respectively, for a total of 28 architects) and solicited feedback via email surveys and an online chat forum.

Interview and focus group data were recorded and transcribed. We analyzed these transcripts, along with notes taken during these encounters, and other feedback provided during the early adopter program (via email surveys and online chat) using thematic techniques [1].

3 System Actions – Making Sense of Smart Actions

The system's initial starting activity is the identification of the customers' IT service requirements. As described above, customers' requirements are typically provided via extensive RFP documentation. Solution designers upload these files into ITDezigner, which then extracts and classifies the RFP text according to a defined list of IT services categories. For example, a requirement text could say: "*Service Provider will add, change, delete, or revoke End User IDs that access applications controlled by Client, per the established security standards.*" The outcome of the system's NLP processing then classifies these text segments using a list of defined IT services – a list of categories like "Account Management," "End User," and "Security" (nearly twenty in total). These service categories are then carried forward in the tool and used as inputs into downstream modeling (optimization modeling that aids the architect in designing a technical solution to cover the requested services, which we will discuss in further detail below).

After extraction, the extracted and classified content must be verified – this is a two-step process that involves assessing both the precision and recall of the system's NLP. One part of this verification process involves inspecting the documents using an HTML viewer, skimming for "white space" (text that was not classified) and manually labelling such content as appropriate. A design mockup of the HTML viewer can be seen in **Fig. 1**.

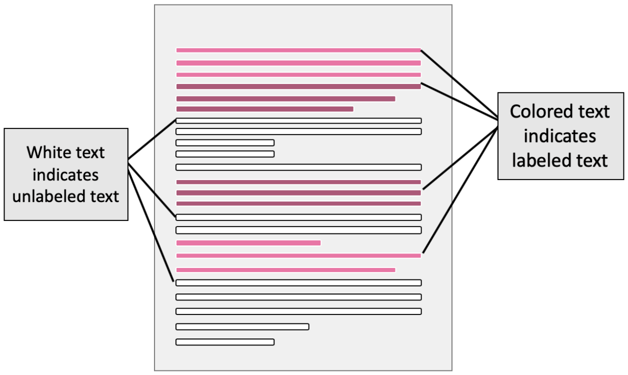

White text indicates unlabeled text
Colored text indicates labeled text

Figure 1. Architects can skim processed documents to verify classified requirements (colored text) and find "white space" to manually classify (design mockup to maintain confidentiality).

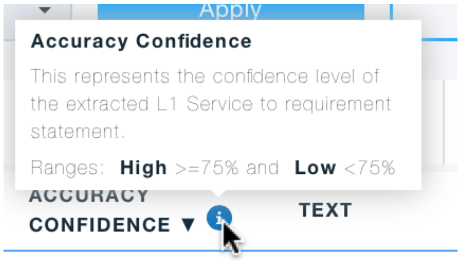
This part of the new process introduced confusion over the meaning of the white space – and what it was communicating to the user. Does white space mean the NLP does not think that text is important? Nick, a technical architect, explained his thought process as he interacted with an RFP's white space:

> To me, (reads RFP text) '*Service Provider will give expert advice with respect to available hardware and software to meet client needs.*' To me, that is a requirement. But that's SOP (meaning Standard Operating Procedure). That's why I wouldn't necessarily pay as much of attention to that... It's standard operating procedure for us to do type of perform... So that's what I mean, I think that's why [the tool] didn't label that. – Nick, technical architect

During usability testing, solution designers wondered if all the white space in a document would need to be manually labelled. Later during the field study this point was clarified during a Q&A session by one of the system's product owners (PO), who explained that the NLP modeling was designed in such a way that each and every requirement did not necessarily need to be classified for the tool to be able to recommend an optimal solution. Once a particular service category (e.g., "End User") was identified in the document, the NLP did not need to extract any other examples of "End User" for its downstream processing to retain its accuracy. An ongoing issue remained, though, on how to best explain the meaning of the white space and how the identified requirements were used downstream in simple and easy-to-understand manner for users (particularly new users) to understand.

Another part of the verification process is reviewing the system's classified requirements for their accuracy – this can be done using a table view within the tool or downloading the extracted requirements in a .csv file. In feedback provided on early prototypes of the system, users expressed a desire for the system to indicate to them where they should focus their verification efforts – it is not uncommon for the system to return hundreds (if not thousands) of classified requirements. Without guidance from the system on what requirements might need more attention than others, solution designers had concerns over the time and effort that might be needed to review each and every requirement. In response to this feedback, later prototypes of the tool included a "confidence rating" for each requirement, allowing users to sort classified requirements based on the system's "High" or "Low" confidence in their accuracy. These accuracy metrics were explained to user's with a hover-over information button, as seen in **Figure 2**.

But the introduction of the confidence metric opened up new questions for architects. As architects interacted with ITDezigner, questions emerged over whether the tool's statistical modelling – the machine intelligence inside the tool trained on hundreds of pages of RFPs from other clients and other deals – would be able to understand the context of the specific client and specific deal at hand. How will I know if the tool has missed something? How will I know if it can't really handle *this* RFP? One of the technical solution managers, Francesca, raised this point. In testing the tool, she uploaded RFP documents from a recent deal her team solutioned. "*It was an unusual deal*," she explained, with the client asking for novel service arrangements her team had never worked on before. She used this as an example to elaborate on her concerns. The tool's NLP models historical data, but a key part of solutioning work is being able to adapt to new demands. "*We can't just have in mind what we've done in the past*," she said. If we only look to RFPs we have responded to in the past and IT solutions we have designed in the past, how do we adapt to new market trends? "*This is an important issue in thinking about how we will use [ITDezigner] in the future*," she said. If the tool cannot handle something new, something that it has not seen before, she felt it needed to make that clear to the users.



Accuracy Confidence

This represents the confidence level of the extracted L1 Service to requirement statement.

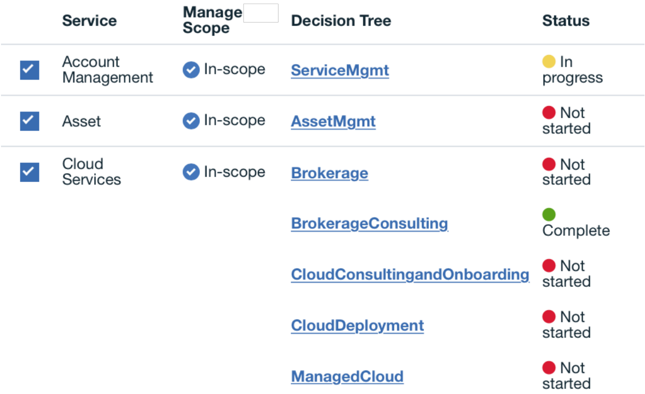
Ranges: **High** >=75% and **Low** <75%

Figure 2. Shows a detail of an explanation of the "Confidence Score" indicator inside the tool.

While this confidence feature provided information valuable in assessing the algorithmic inner-workings of the NLP, open questions still remained. How do I use this score? Is low confidence indicating something strange or out of the ordinary with this text, something the NLP model hasn't seen before (with the implication being that all downstream processing in the system would continue to carry this low confidence in the solutions it proposed)? Or is this low confidence more a function of the nature of NLP, difficulties inherent in algorithmically classifying text that we would expect to see even in RFPs that are a "good fit" for us? What these issues surface is a disconnect between the underlying logics of NLP techniques – creating a "global" model of text fragments from a training dataset – and the more "local" specifics of a given RFP. One perspective is fundamentally backward-looking, the other forward-looking. When the algorithm tells me it has low confidence in its classifications, how do I use that information in building the solution for this particular RFP? Francesca wasn't sure, but felt this was a critical question solution designers would need to be clear on when figuring out how to use the tool's outputs in their solutioning work.

4 Interactivity – Understanding Impact of User Actions on System Outputs

After the requirements have been identified for an RFP, the next step in the process is to compose and optimize a design solution that satisfies the customer's requested technical requirements. Taking as input the list of IT service categories derived from the process we described above, the system makes a preliminary recommendation of suggested offerings to include in the solution. Preliminary recommendations can then be tailored using an input feature inside the system referred to as "Decisions Trees." These trees are composed of a series of questions, specific to each offering in the company's catalog, which guide the user in selecting additional (or alternative) offerings to add to the preliminary set the system has suggested. Activities inside the decision trees are reflected in the UI as either "Not Started," "In Progress" or "Complete" (see **Figure 3**).



	Service	Manage Scope	Decision Tree	Status
☑	Account Management	✓ In-scope	ServiceMgmt	● In progress
☑	Asset	✓ In-scope	AssetMgmt	● Not started
☑	Cloud Services	✓ In-scope	Brokerage	● Not started
			BrokerageConsulting	● Complete
			CloudConsultingandOnboarding	● Not started
			CloudDeployment	● Not started
			ManagedCloud	● Not started

Figure 3. Shows a detail of the tool's Solution Builder feature, where the status of Decision Trees are displayed as "Not Started," "In Progress," or "Complete."

The decision tree include a number of detailed questions about technical specifications and other customer information and these status indications can help solution designers keep track of their progress over the course of multiple sessions. While these indicators are helpful in keeping one's place, there is a lack of clarity on how each response inside the decision trees will impact the offering selection and ultimately, the overall solution. While the users understood that their responses to the questions had some type of impact on the later solutions recommended by the system, they were not sure exactly the relationship (i.e. which specific responses they chose led to which of the tool's modified recommendations). Solution designers wanted the ability to understand this type inter-active coupling of their actions and subsequent impact on the systems outputs.

The system has an "Explain" feature, spreadsheets can be downloaded that outline the offerings coverage for a given solution. These spreadsheets involve a x,y axis with IT services on one and the corresponding offerings that cover those services on the other. While this is useful, it is not dynamic and isn't able to support running "if/then" type scenarios, where architects can more clearly make sense of the impacts of different offering selections. There is a difference between providing a report that explains what is covered at a given time, and one that explains how the user's actions impacted that coverage. Such if/then scenarios are important to support solution designer's understanding of what is happening (sensemaking), but also ensure the proposed solution's coverage is really "optimal" (and they are able to justify offering selections to others within their team).

5 Organizational Processes – Integrating System Outputs With Established Practices

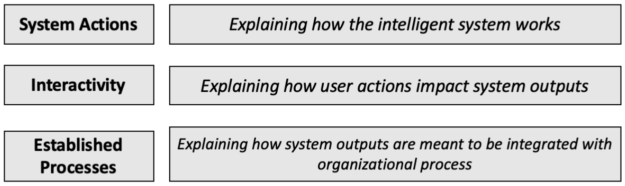
In addition to explanations of the system's inner workings and how their actions affected optimization of the solution, solution designers also expressed the need to understand how the system – and particularly the various outputs it was capable of producing – were meant to be incorporated into the existing workflow processes for IT service architecture design. These concerns presented a practical problem: *"How do you get that stuff from the tool, which is great, into the documents that people are expected to use today for auditing and recordkeeping?"* Angie, a technical solution manager, wondered. *"I was not sure how to do that, other than some kind of weird manual cut and paste right?"* When new systems are introduced into an existing work practices, workers must figure out how to integrate novel artifacts produced from novel systems with existing artifacts and existing systems. But workers must also figure out how to align novel systems and artifacts with the broader organizational processes they implicate. *"We've got to think through how does that all fit with what is a fairly industrialized process,"* Nathan, a technical solution manager, said. *"The minute you've got a process and it's understood by a lot of people to work a particular way, if you want to change it,"* he explained, *"somebody would have to articulate the changes and tell people what they can and cannot do and are supposed to do especially based on what they couldn't do before or did do before. That would be the best to me to understand that linkage."* Such comments point to the need for explanations of a different sort – in addition to explanations of the system's technical actions and explanations of how use of the system influences outputs, solution designers also were searching for explanations of the intended alignment of the intelligent system with the broader contextual reality of organizational life. Through everyday work practices, the intelligent system and the outputs it produces will figure into complex logistical workflows that include, but also exceed any individual workers' efforts – and explanations are needed to understand what shape these alliances are intended to take.

Another part of successfully integrating the system's outputs with established protocol was the issue of partiality; what to do with partial outputs. An example of this was in the tool's optimization piece that takes extracted IT requirements and matches them to the company's catalog of services. *"So what this tells me,"* Stacey, a technical solution manager, wondered aloud as she inspected the interface, *"is that 70% of the solution is covered with standard offerings from the catalog...is the other 30% something that I would have to then go and generate something on the side to get the custom pieces?"* This concern highlights the importance of hybridity in the roll out of intelligent workplace tools – alongside various technical explanations of smart systems, workers will also need to understand what is needed to bridge the gap between the "old" and "new" ways of working such systems are meant to support.

6 Summary

In this position paper, we have described issues around explanation that have surfaced from an ongoing study of the design, development, and deployment of an intelligent workplace system in the IT services domain. These issues center on three themes – system actions, interactivity and system outputs, and integration with existing processes. In each, different types of explanations are sought by users – system actions involves making sense of and assessing the accuracy of system actions. Interactivity and system outputs focuses on users' abilities to identify and understand the impact of their actions on system outputs. Integration with existing processes focuses on users' understanding of the system's outputs in relation to established organizational processes.

Our initial findings point to the need to more carefully articulate the different types of explanations needed to support workers as they make sense of and successfully integrate smart systems in their everyday work practice. We outline these initial explanatory concerns in **Figure 4**, which contributes to the workshop's themes of defining explanation, as well as the UX design and placement of explanations at differing points in the user's experience. By participating in the ExSS Workshop at IUI'19 we hope to contribute to the workshop's discussion by providing an industrial intelligent system case study and by outlining our initial analysis of how and why explanations are an important issue in this project. As our project is ongoing, we also hope to discuss design approaches to address the issues we have raised here, as well as our future work on understanding the contours of smart system explainability in situated work contexts.



System Actions	*Explaining how the intelligent system works*
Interactivity	*Explaining how user actions impact system outputs*
Established Processes	*Explaining how system outputs are meant to be integrated with organizational process*

Figure 4. Outlines Different Types of Explanation Described in Our Paper.

ACKNOWLEDGMENTS

Thank you to the IT solution designers who graciously shared their time and insights during the field study. All opinions are our own and do not represent any institutional endorsement.

REFERENCES

[1] Braun, V. and Clarke, V. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology*. 3, 2 (Jan. 2006), 77–101. DOI:https://doi.org/10.1191/1478088706qp063oa.

[2] Kuang, C. 2017. Can A.I. Be Taught to Explain Itself? *The New York Times*.

[3] Pearl, J. and Mackenzie, D. 2018. AI Can't Reason Why. *Wall Street Journal*.

[4] Someren, M.W. van et al. 1994. *The think aloud method: a practical guide to modelling cognitive processes*. Academic Press.

[5] Wolf, C.T. and Blomberg, J.L. 2019. Intelligent Systems in Everyday Work Practices: Integrations and Sociotechnical Calibrations. *Intelligent Human Systems Integration 2019* (2019), 546–550.

[6] 2017. Editorial: Ceding powers of decision to AI presents a paradox. *Financial Times*.