6-23-2022

Information Flow Optimization in Augmented Reality Systems for Production & Manufacturing

Mahdi Moghaddam

Shimon Y. Nof

Information Flow Optimization in Augmented Reality Systems for Production & Manufacturing

Moghaddam, M., and Nof, S.Y.

PRISM Center and School of IE, Purdue University

PRISM Research Memorandum No. 2022-P3

June 23, 2022

PRISM Lab/Purdue

PRISM Center
Production, Robotics, and Integration
Software for Mfg. & Management

"Knowledge through information; Wisdom through collaboration"

PRISM Center
Grissom Hall
315 N. Grant Street
Purdue University
West Lafayette, IN 47907-2023

Information Flow Optimization in Augmented Reality Systems for Production & Manufacturing

Mahdi Moghaddam, Shimon Y. Nof
PRISM Center & School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA
{mmoghad, nof}@purdue.edu

Keywords: Automated AR, CCT Collaborative Control Theory, CI Collaborative Intelligence, Future Factories, Future Work, HUB-CI Protocol

Abstract: As a part of our research to optimize the collaborative work and factories of the future, augmenting human abilities and skills, and enabling collaborative automation, we have designed a framework which optimizes augmented reality (AR) systems in production and manufacturing. This framework consists of protocols and modules that control the flow of information delivered to and from workers. The protocol to which most of this article is dedicated is a HUB-CI (hub for collaborative intelligence) protocol that prioritizes and selects the most relevant information to a worker's unique characteristics and current activity. Since 2008, different types of HUB-CI models have been developed, implemented, and refined by the PRISM Center at Purdue University. In general, HUB-CI serves as a cyber-collaborative controller distributed in a system's control points, where material, data, information, and decision flows converge and must be distributed to points of use. We call such control points *Flow Junctions (FJ)*. A FJ is defined within this framework and the HUB-CI protocol designed for it is evaluated in a simulation model of a manufacturing assembly task.

1 INTRODUCTION

Augmented reality (AR) is one of the technologies that can transform production and manufacturing industries. It is the augmentation of the real, physical world through digital content, where virtual objects are superimposed on the real environment (Azuma, 1997). As a human-machine interaction tool, AR is expected to help industries cope with the increasing complexity of their human-in-the-loop workflows and processes (Ong et al., 2008). These new workflows demand the collaboration of workers with intelligent systems and machines and adaptation to dynamic changes in task structures. The ability of all involved agents to collaborate is defined as collaborative intelligence (CI) (Devadasan et al., 2013; Zhong et al., 2015b, 2015a). We optimize CI of agents in collaborative workflow by controlling data and information flow between them. Therefore, there are control points, which we call *Flow Junctions (FJ)*, in which data, information, and intelligence controllers must be implemented. Such controllers are known as *HUB-CI* (hub for collaborative intelligence) (Dusadeerungsikul et al., 2019; Nair et al., 2019; Seok & Nof, 2011; Zhong et al., 2013, 2014; Zhong & Nof, 2013).

In the context of AR, an FJ is situated between experts, software systems, etc. and the AR display of the worker. In the normal workflow of production and manufacturing, workers interact with machines and processes while performing tasks that are designed by workflow designers and experts, which require data and information exchange. AR serves as a medium for this exchange to facilitate interactions and augment workers' performance. Due to the ever-present access of AR system to a worker's field of view, however, delivering information to the worker through AR cannot be unrestricted. The virtual objects of AR, which we refer to as *AR elements*, require their own processing time and add an additional mental workload on the worker. Therefore, we must identify what information to deliver and at what time.

A HUB-CI protocol optimizes the flow of information through the FJ between AR and workflow. The objective of this article is to develop this HUB-CI protocol, i.e., to maximize the added value of information sent through FJ to the AR display and subsequently to the worker. We aim to achieve this objective by answering the following research questions:

- ***RQ1:*** How can we prioritize information based on its relevance to the current state and knowledge needs of the worker?
- ***RQ2:*** How can we personalize AR for every worker based on their attributes, experience, and performance?
- ***RQ2:*** How can we optimize the timing of information flow and knowledge delivery?

Applications of AR in manufacturing include manual assembly, robot programming and operations, maintenance, process monitoring, training, quality inspection, picking process (de Souza Cardoso et al., 2020). The HUB-CI protocol can be implemented in all these applications. Moreover, we recognize current limitations of AR, which are mostly related to hardware as identified by researchers (de Souza Cardoso et al., 2020). Therefore, the models that we present in this article are aimed at future AR systems in which such limitations have been and are being rectified.

The AR design framework based on the presented HUB-CI protocol is flexible and open to expansion. It will give researchers and developers a blueprint to follow in developing AR systems that adds a new dimension to existing workflows rather than a simple change in the medium of information exchange, which does not justify the cost of implementing an AR system. Our research objective addresses this shortcoming and aims to enable widespread deployment of AR in industry as complementary augmentation of human workers (Acemoglu, 1998). We acknowledge the necessity of human presence in workflows of future factories and have aligned our contribution in this article with the goal of optimizing such workflows with human workers in mind. The human-complementary and worker-augmentation aspects of AR are enhanced by addition of HUB-CI protocol to the workflow. After answering the research questions defined above, we will have AR systems that are proactive in assisting workers by providing responsive, relevant, and timely information, and adaptive to every worker's unique needs.

2 AR DESIGN FRAMEWORK

The presented AR design framework has been developed based on the objective and research questions defined in Section 1. It is comprised of multiple processes, protocols, and modules that work together to create a dynamic and adaptable AR system for workers of future factories. Our design objective (based on research questions defined above) is that information must be prioritized based on the current activity of workers, their physiological state, information that they are currently receiving, and their unique attributes. Figure 1 illustrates the design framework, and its components of the framework are presented in the following subsections.

2.1 AR Elements

In this framework, we consider an AR element to be an independent, and meaningful expression of a single type of information (textual, numerical, graphical, animation, auditory instructions, etc.) that convey a message to the worker which belongs to one message category. Each AR element has a virtual object, designed by an AR designer based on the task design, and a set of parameters. A manufacturing task designer assigns the values of these parameters based on empirical data, expert instructions, CAD, etc. The parameters of an AR element are as follows.

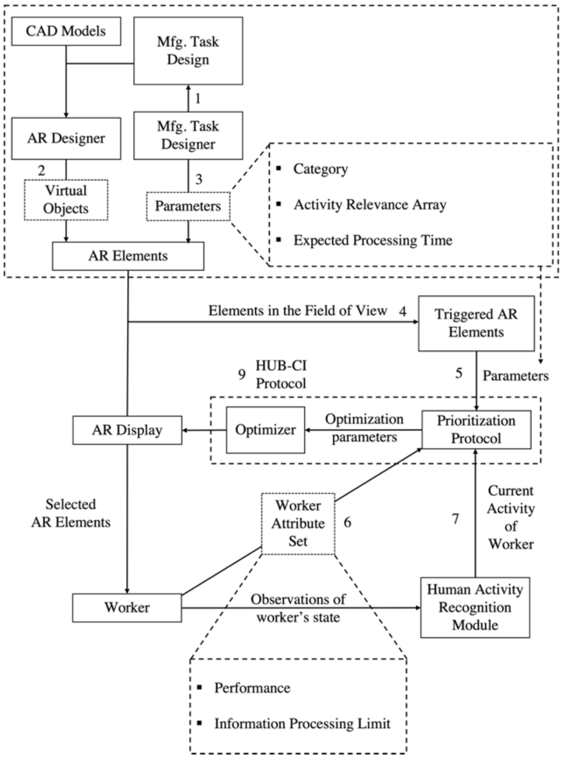


Figure 1: AR design framework with HUB-CI protocol.

2.1.1 Category

Categories include instructions on how to perform a specific task, numerical values or data, process or machine status, notifications, warning for safety, detected errors, and potential conflicts. The categories are identified by a binary array. Assuming that there are k categories, we represent them as $ct = [ct_1, \dots, ct_k]$ where $ct_i \in \{0,1\}$ represents whether the element belongs to the ith category. We assume that in the design of AR elements, an element belongs to only one category.

2.1.2 Activity Relevance Array

Manufacturing tasks follow a specific structure and within that structure we can identify the tasks that the worker is expected to perform. Furthermore, we can use observations of the activities of workers to label activities with finer granularity. The activities can be recognized by a human activity recognition module. We represent the activity relevance array by $ar = [ar_1, \dots, ar_m]$ where $ar_i \in [0,1]$ is the relevance of the AR element to the ith activity. These values are initialized by an expert (e.g., manufacturing task designer) and are heuristically updated over time.

2.1.2 Expected Processing Time

As mentioned in the introduction, the amount of information or number of AR elements presented to the worker requires a limit, as they add cognitive workload and distraction. This limit can be established in a variety of ways, depending on the tasks, workflow, AR elements, predictive behaviors, and what metrics are accessible. For instance, we can use working memory capacity of workers as a limit for how much information we deliver to them. Working memory is a form of memory that can hold a limited amount of information for a short period of time to be processed in the prefrontal cortex as control center or “central executive”, responsible for functions such as decision making and problem solving (Chai et al., 2018; Knudsen, 2007). However, we would also have to measure and predict how much working memory an AR element would require, which is easier said than done.

An alternative metric is time. Workers require some time to process the information an AR element contains, which can be measured through experimentation. On the other hand, there are time constraints for completing activities and tasks to keep up with deadlines. Therefore, a limited time can be allocated for receiving or interacting with AR elements within a period, which will be used later as a parameter in the optimization problem. We denote the time a worker needs to process an AR element as t_i.

2.2 Triggered AR Elements

A subset of the set of AR elements, these are the elements that are triggered when they are in the field of view and attention of worker; when the workflow process reaches a specific stage; or when we wish to alert the worker about a change in the environment. The triggered elements must *compete* for the worker’s attention, i.e., be activated in the AR system. The elements are selected based on their parameters, which were described earlier, and a worker’s attribute set. The parameters of the triggered AR elements will be sent to the Prioritization Protocol.

2.3 Worker Attribute Set

In this framework, personalization occurs through the information retrieved from a worker’s attribute set, which represent the unique skills, experience, and other characteristics of the worker. Each set contains:

1. *Information Processing Limit:* The ability and time required to process information varies among workers. The factors involved are beyond the scope of this article. However, we can anticipate that a worker may process information quicker or slower than the expected processing time measured for a specific AR element. Time of day/week, fatigue, and other factors can also influence this ability of workers. Measuring these variables in real-time can give a more accurate real-time expectation of workers’ information processing capabilities (Meijman, 1997). Considering this characteristic of workers makes the AR system more adaptive.

2. *Performance:* The performance of every worker is measured through their task completion and error rates. A worker’s performance on a certain task or activity will affect the prioritization for the category of AR elements. Recall that we assumed there are k categories of AR elements. The value $pr_i \in [0,1]$ represents the relationship between the worker’s performance and the ith category of AR element for one activity or task. All k values are contained in vector $pr = [pr_1, \dots, pr_k]$. Note that the initial values are assigned by an expert (e.g., manufacturing task designer) and will be updated over time through data collection and analysis.

2.4 Human Activity Recognition & Prediction Module

One of the insights we can gain from the abundant data and data collection methods is to allow automated systems to understand what a human-in-the-loop, in this case manufacturing worker, is doing now and will be doing in near future. This is known as human activity recognition and prediction, which has applications in human-robot interactions (HRI), human-robot collaboration (HRC), surveillance systems, manual workflow analysis, etc. (Bulling et al., 2014; Lasota et al., 2017; Li & Fu, 2014). Even though human activity recognition and prediction are beyond the scope of this article, it is noteworthy that they are essential parts of a collaborative human-in-the-loop system’s CI. Just as workers need to be aware of the system’s state including robot activities, machine and process status, and workflow procedures, by creating mental models and anticipating future states, the automated part of the systems should also be aware of the worker’s current and future activities or intentions. Therefore, an automated AR system must be aware of its user’s state, including activities and intentions.

A module in this framework is responsible for collecting observation data on the state of workers, such as pose, gestures, location in the workspace, gaze, and objects they interact with. It will use this data and task structure to recognize and predict current and future activities of workers (Li & Fu, 2014). As mentioned before, the implementation of this module is beyond the scope of this article, therefore, we assume that we already know what a worker is doing now and will be doing in next.

2.5 HUB-CI Protocol

The HUB-CI protocol of this framework was introduced above in Section 1, as the controller of data and information. The role of this protocol is to select triggered AR elements that are most relevant to (1) worker's current activity, (2) worker's unique attributes (characteristics), and (3) events in the worker's environment. Therefore, HUB-CI protocol consists of two steps: prioritization and optimization. It is noteworthy that these objectives can be expanded depending on particular requirements of each system.

2.5.1 Prioritization

Following the three objectives defined above, HUB-CI prioritizes the triggered AR elements. The steps are as follows.

1. First, HUB-CI must determine the priority of the element given its category and the worker's performance, which is $p^1 = ct \times pr$.
2. Then, it receives the current activity of the worker from human activity recognition module. Assuming that the worker is performing the jth activity, HUB-CI will receive a vector ac whose jth element is 1 and the rest are 0. We can show the relevance of an AR element to worker's activity by $p^2 = ac \times ar$.
3. Steps 1 and 2 provide two numerical values between 0 and 1. The multiplication of these values determines the priority of the AR element. The intuition behind it is that both values can serve as independent priorities, thus when multiplied together serve as relative weights for each other. The priority of the ith triggered AR element is

$$p_i = \left(\begin{bmatrix} ct_1 \\ ct_2 \\ \vdots \\ ct_k \end{bmatrix} \begin{bmatrix} pr_1 \\ pr_2 \\ \vdots \\ pr_k \end{bmatrix} \right) \times \left(\begin{bmatrix} ac_1 \\ ac_2 \\ \vdots \\ ac_m \end{bmatrix} \begin{bmatrix} ar_1 \\ ar_2 \\ \vdots \\ ar_m \end{bmatrix} \right) = p_i^1 p_i^2 \quad (1)$$

2.5.1 Optimization

Using the priorities defined above, we formulate the optimization problem as a 0/1 knapsack problem. Our objective is to select AR elements that have the highest priority (relevance and thus added value). We represent the decision variables as a vector $x = [x_1, \dots, x_n]$, where x_i is a binary variable representing whether element i is selected or not. The priority and required processing time of each element are received from the prioritization module as $p = [p_1, \dots, p_n]$ and $t = [t_1, \dots, t_n]$, where p_i and t_i are defined in Section 2.5.1 and 2.1.2 respectively. We denote the capacity of the knapsack as t_c, which is defined in Section 2.3. Thus, our optimization problem reads

$$\underset{x_i}{\text{maximize}} \quad \sum_{i=1}^{n} p_i x_i \qquad (2)$$

$$\text{subject to} \quad \sum_{i=1}^{n} t_i x_i \leq t_c \qquad (3)$$

$$x_i \in \{0,1\}, \quad \forall x_i \in x$$

3 EXPERIMENTS

We have designed a discrete-event simulation model of an assembly task as a proof of concept. In this model, there are four workers with different level of skills and performance: (1) expert, (2) experienced, (3) novice, and (4) trainee. In parallel to the four performance levels, we have simulated four levels of expressed instructions, which are assumed to be designed as AR elements. The simulated assembly task consists of two sub-tasks, each of which consists of two more sub-tasks, which can be performed by four activities. Figure 2 shows this hierarchy.

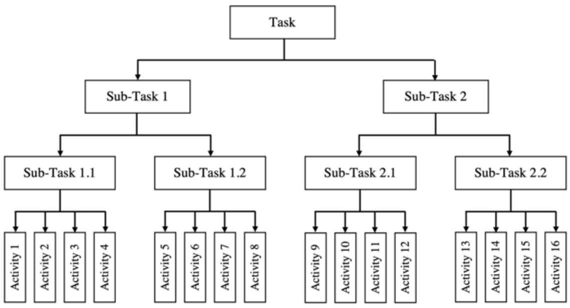


Figure 2: hierarchy of the task, sub-tasks, and activities in simulation model.

The simulation loops over events, where an event is the completion of an activity by a worker and the simultaneous changes that occur in the state of environment and worker. During the simulation we already know the next activity of the worker, but in real-world implementation of this framework, the next activity is predicted by the human activity prediction module. Apart from instructions, these changes are also communicated to the worker as machine or process status, warnings, and notifications by their respective AR elements. Figure 3 shows the simulation process.

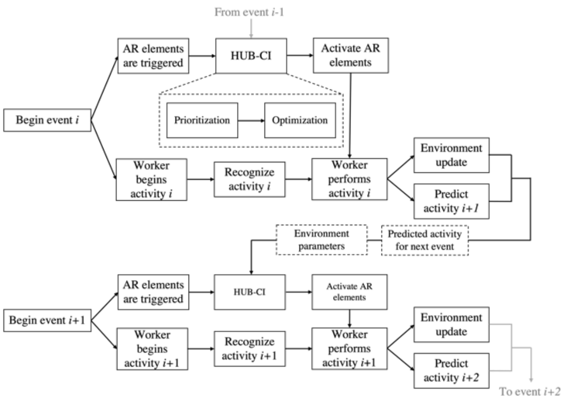


Figure 3: simulation process based on discrete events.

3.1 Simulated Triggered AR Elements

Each simulated AR element contains: (1) content, (2) category, (3) activity relevance array with values between 0 and 1, (4) expected processing time, which indicates how much time (in seconds) it takes for the worker to view the element and understand its content. Elements are triggered by either coming into worker's field of view or when the assembly process reaches a specific stage or when a change occurs in the workspace about which the worker must be notified. AR elements simulated in this case study are as follows.

3.1.1 Simulated Instructions

Each worker performs sixteen activities to complete the task. Note that this number could be arbitrarily larger, but we selected the minimum representative number of iterations that would cover different types of changes in the environment, hence different categories of AR elements. There are instructions expressed for every level of subtask and activity, which are shown to the workers based on their performance. For instance, an expert worker is expected to receive one general instruction about the entire task, which will be available until the task is completed; while at the other level extreme, a trainee will receive instructions expressed per activity.

In real-world implementation, however, the performance level of workers will vary depending on the activities they perform. Therefore, we have assumed that based on the performance of the experienced and novice workers, they need to receive more detailed instructions expressed for certain activities and implemented it in the simulation. Furthermore, if the AR element of an instruction is displayed to a worker and would be active across multiple events, its processing time will be reduced to minimum by the worker's learning and increasing level of experience.

3.1.2 Status, Warnings, and Notifications

There are three different categories of AR elements in addition to instructions in this simulation. We have assumed that in some of the activities the worker is operating with a machine (e.g., lathe, collaborative robot). The status of the machine is designed as an AR element and its relevance to each activity is displayed by the designer in the activity relevance array.

Similarly, we have added warnings (alerts) of various types including mobile robot path entering worker's workspace, collaborative robot's motion, and faulty part detected. The activity relevance array of warnings is filled based on the event in which they occur. For instance, a mobile robot is close to the worker in event j to j+3, during which the value of its relevance is high. Note that this example is only applicable to this simulation model; in real-world scenario, such warnings are conveyed to the worker as soon as they are detected, irrespective of the activity or relevance to this worker.

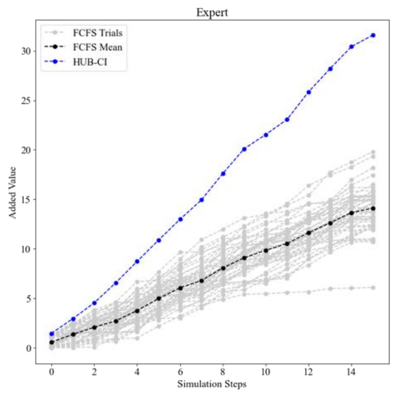
The third category of AR elements is notifications. The difference between notifications and warnings is that notifications do not have the same urgency. They are designed to help workers have a better understanding of the present and future states of the process and themselves. The notifications that we included in the simulation include worker's physical state (fatigue, heartrate, continuous working hours, etc.) and expected task completion time. Recall that the relevance of these AR elements to the worker's current activity is subjectively determined by the designer and heuristically and continuously improved.

3.1.2 Prioritization, Optimization, and Results

The equation defined in Section 2.5.1 is used to determine the priority of the triggered AR elements. This step will produce an array of priority values that are between 0 and 1. Subsequently, we can extract the required processing time of each element and add them to another array. As described in Section 2.1.2, the capacity used in the optimization problem is the maximum time we can allow the workers to observe or interact with AR elements while keeping up with the expected task requirements and their completion time. Subsequently, we formulate this selection problem as a 0/1 knapsack problem where priority and required processing time arrays serve as profit and weight, respectively. After solving the optimization problem, the selected elements are activated (expressed). These steps are repeated for every event.

We have compared this framework and results of HUB-CI protocol to a first-come-first-serve (FCFS) approach in which given the time constraint, AR elements are expressed as soon as they are triggered. There is obviously no guarantee that the first element to be triggered is the optimal. On the other hand, if we activated all the triggered elements, we would be giving data and information to workers, that are not necessarily relevant to their attribute set, or current activity. The results of the first analysis are shown in Figure 4, where we compare the added value of this framework relative to the FCFS approach. Note that the added value is the sum of the values selected from the priority array. The improvement of HUB-CI protocol on FCFS is statistically significant across all worker experiences. Table 1 shows the p-values of t-test and one-way ANOVA.

The reader may notice the similarity between the lines of added value for HUB-CI for all workers. This similarity occurs because the maximum value of any instruction's relevance for all workers is 1, but it does not mean that they receive the same instructions. To illustrate this, we have calculated the added value of the AR elements selected for the trainee when shown to other workers. The results of the comparison are shown in Figure 5.



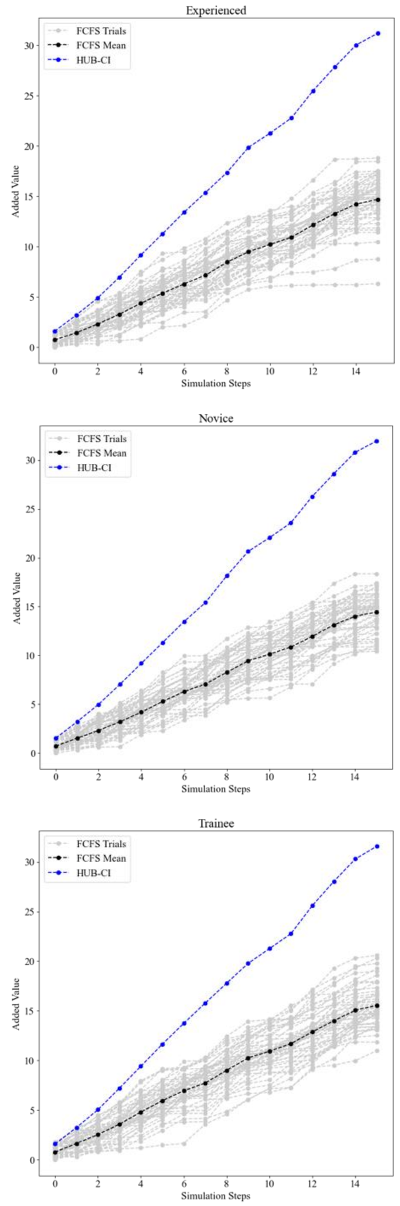


Figure 4: added value of HUB-CI protocol vs. fifty trials of FCFS protocol for four types of workers.

Table 1: p-values of t-test and one-way ANOVA for HUB-CI results and FCFS average.

Workers	p-value	
	t-test	One-way ANOVA
Expert	0.0026	0.0026
Experienced	0.0032	0.0032
Novice	0.0024	0.0024
Trainee	0.0049	0.0049

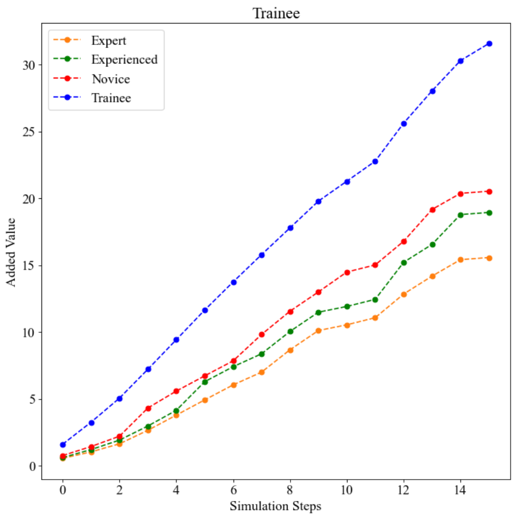


Figure 5: comparison of added value of trainee's instructions for other workers.

4 CONCLUSION & FUTURE STEPS

One of the common criticisms of frameworks such as the one presented in this article is their implementation in real-world systems. Therefore, we would like to acknowledge that this framework, particularly HUB-CI protocol, can be implemented in any AR system and manufacturing workflow just as it was implemented in the simulation model process explained in Section 3. Furthermore, it is possible to expand and tailor the presented framework according to the requirements of each system.

One of the limitations of this work is the lack of access to real-world data and workflows for evaluating the framework. Even though we have made reasonable assumptions in our experiments, real data enforces the robustness of the framework against anomalous scenarios, errors, and exceptions. Moreover, we would like to implement this framework in an AR system designed for production and manufacturing tasks.

ACKNOWLEDGEMENTS

The purpose of this article is to study the tools for enabling human-in-the-loop collaborative automation for cyber collaborative future work and factories (C2F) as part of our NSF Grant 1839971: Collaborative Research: Pre-Skilling Workers, Understanding Labour Force Implications and Designing Future Factory Human-Robot Workflows Using Physical Simulation Platform. This research has also been submitted to ICPR-AR 2022, Curitiba, Brazil, November 2022.

REFERENCES

Acemoglu, D. (1998). Why do new technologies complement skills? Directed technical change and wage inequality. *Quarterly Journal of Economics*, *113*(4). https://doi.org/10.1162/003355398555838

Azuma, R. T. (1997). A Survey of Augmented Reality. In *Presence: Teleoperators and Virtual Environments* (Vol. 6). http://www.cs.unc.edu/~azumaW:

Bulling, A., Blanke, U., & Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, *46*(3). https://doi.org/10.1145/2499621

Chai, W. J., Abd Hamid, A. I., & Abdullah, J. M. (2018). Working memory from the psychological and neurosciences perspectives: A review. In *Frontiers in Psychology* (Vol. 9, Issue MAR). https://doi.org/10.3389/fpsyg.2018.00401

de Souza Cardoso, L. F., Mariano, F. C. M. Q., & Zorzal, E. R. (2020). A survey of industrial augmented reality. *Computers and Industrial Engineering*, *139*. https://doi.org/10.1016/j.cie.2019.106159

Devadasan, P., Zhong, H., & Nof, S. Y. (2013). Collaborative intelligence in knowledge-based service planning. *Expert Systems with Applications*, *40*(17), 6778–6787.

Dusadeerungsikul, P. O., Sreeram, M., He, X., Nair, A., Ramani, K., Quinn, A. J., & Nof, S. Y. (2019). Collaboration requirement planning protocol for HUB-CI in factories of the future. *Procedia Manufacturing*, *39*, 218–225.

Knudsen, E. I. (2007). Fundamental components of attention. In *Annual Review of Neuroscience* (Vol. 30). https://doi.org/10.1146/annurev.neuro.30.051606.094256

Lasota, P. A., Fong, T., & Shah, J. A. (2017). A Survey of Methods for Safe Human-Robot Interaction. *Foundations and*

Trends in Robotics, *5*(3). https://doi.org/10.1561/2300000052

Li, K., & Fu, Y. (2014). Prediction of human activity by discovering temporal sequence patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(8). https://doi.org/10.1109/TPAMI.2013.2297321

Meijman, T. F. (1997). Mental fatigue and the efficiency of information processing in relation to work times. *International Journal of Industrial Ergonomics*, *20*(1). https://doi.org/10.1016/S0169-8141(96)00029-7

Nair, A. S., Bechar, A., Tao, Y., & Nof, S. Y. (2019). The HUB-CI Model for Telerobotics in Greenhouse Monitoring. *Procedia Manufacturing*, *39*, 414–421.

Ong, S. K., Yuan, M. L., & Nee, A. Y. C. (2008). Augmented reality applications in manufacturing: A survey. *International Journal of Production Research*, *46*(10). https://doi.org/10.1080/00207540601064773

Seok, H., & Nof, S. Y. (2011). The HUB-CI initiative for cultural, education and training, and healthcare networks. *21st International Conference on Production Research: Innovation in Product and Production, ICPR 2011 - Conference Proceedings*.

Zhong, H., Levalle, R. R., Moghaddam, M., & Nof, S. Y. (2015a). Collaborative intelligence - Definition and measured impacts on Internetworked e-Work. *Management and Production Engineering Review*, *6*(1). https://doi.org/10.1515/mper-2015-0009

Zhong, H., Levalle, R. R., Moghaddam, M., & Nof, S. Y. (2015b). Collaborative intelligence-definition and measured impacts on internetworked e-work. *Management and Production Engineering Review*.

Zhong, H., & Nof, S. (2013). Collaborative design for assembly: the HUB-CI model. *22nd Int. Conf. Production Research*.

Zhong, H., Wachs, J. P., & Nof, S. Y. (2013). HUB-CI model for collaborative telerobotics in manufacturing. *IFAC Proceedings Volumes*, *46*(7), 63–68.

Zhong, H., Wachs, J. P., & Nof, S. Y. (2014). Telerobot-enabled HUB-CI model for collaborative lifecycle management of design and prototyping. *Computers in Industry*, *65*(4), 550–562.