A Web-based Mixed Reality Interface Facilitating Explicit Agent-oriented Interactions for Human-Robot Collaboration

Joe David
Faculty of Engineering and Natural Sciences
Tampere University
Tampere, Finland,
Faculty of Engineering
Norwegian University of Science and Technology
Trodheim, Norway
joe.david@tuni.fi, joesd@stud.ntnu.no

Eeva Järvenpää
Faculty of Engineering and Natural Sciences
Tampere University
Tampere, Finland,
eeva.jarvenpaa@tuni.fi

Andrei Lobov
Faculty of Engineering
Norwegian University of Science and Technology
Trodheim, Norway
andrei.lobov@ntnu.no

***Abstract*— Communicating and recognizing intent is a crucial part of human-robot collaboration (HRC). It prevents competing and turn-taking behaviour that would otherwise lead to inefficient and unsafe collaborative activities. This study presents a novel approach towards enabling standards-based explicit bi-directional intent communication. The approach entails projecting a tailored web-based user interface upon the worktable shared between a human and robot agent. The interface has a close integration with an agent framework (JADE) that allows intent communication via mechanisms standardized by the IEEE Computer Society. The interaction model is discussed for its rationale and the possibilities it exposes as future work.**

Keywords—human-machine interface, human-robot collaboration, mixed-reality, agent-based systems, intent

I. Introduction

Collaborative relationships between robots and humans have been in hot pursuit after the highly successful third industrial revolution that led to automation with autonomous and isolated robots [1]. The motivation is the amalgamation of complementary skills possessed by human and machine to achieve a higher overall productivity and better product quality [2]. Such collaborative environments position human-machine interaction into a cyber-physical environment as part of the fourth industrial revolution [2][3].

A multi-agent view rightfully seems to be the most appropriate view of humans and robots in an HRC environment [4]. The agent view of humans and robots assumes autonomous behaviour on part of both agents. An agent needs to exhibit responsibility and leadership when performing tasks and assume either a leading or supportive role with respect to the collaborative task. Role selection is one of many facets that must be communicated as intent. What else must be communicated is application dependent. For example, the intended path of an agent might be another that's relevant in a mobile-robot [5] or even an articulated robot use-case [6]. Further, roles that agents take and its selection must be assigned dynamically at run-time and must not be a static one-time activity [1]. However, most of recent research assumes fixed agent roles pre-determined before task execution [2].

This study takes the view that in order to assume roles and responsibilities without competing and turn-taking behaviour, the desire for the same must be communicated as intent between collaborating agents dynamically at run-time. There needs to be an interaction model that facilitates this bi-directional communication (*the how*) and an information model that expresses the entailed semantics of communication (*the what*). Also, collaboration with humans necessitates that this model is comprehensible to humans and reduces their cognitive load [1] whilst being unobtrusive, effective and safe. The question then becomes, as put by Hayes and Scassellat [1], "How can a robot leverage channels of communication that humans understand, despite dissimilar physical forms or capabilities?" The answer to the former part of the question lies in the development of mechanisms of communication, the likes of which is attempted in this paper. The latter is a familiar problem commonly addressed by development information models that the agents use to form relatable mental attitudes and represent capabilities and other forms of existence which is left as future work.

To this end, with the objective to address '*the how*' or the former part of the afore mentioned question, this research employs a web-based user interface that is projected on the worktable of the collaborating agents. Similar projection-based interfaces have in the past been found to be useful in industrial workbenches [7]. Specifically, they have been proven to be advantageous for complex tasks with reduced error rates [7]. We take such an approach further to incorporate bi-directional communication using the said projection-based interface in this paper.

The novelty of the study is threefold. They are (i) establishing bi-directional communication in a collaborative environment between an operator and a robot using only a pure projector-based interface (ii) enabling a communication based on standards for the afore-mentioned purpose and thus the possibility to carry out 'complex conversations' via interaction protocols (iii) the rich interactive experience it opens up possibilities for owing to the use of a mature UI platform, i.e., a web-browser.

The next section reviews literature for existing approaches for communication of intent between humans and robots. Section III sets the research objectives and describes the research setting. Section IV provides background information to complement the understanding of the interaction model presented in Section V. In Section VI, the interaction model and its design rationale are discussed. Section VII summarizes the study and mentions the future direction of the research.

II. Related Work

After a brief review of modalities of explicit and implicit intent recognition and communication in literature we review state of the art of extended reality interfaces for explicit intent communication from which our work draws inspiration.

Speech is a natural means of communication for humans and thus has garnered a lot of interest in explicit intent communication between robots and humans. Today's robots can not only understand speech but also synthesize speech to communicate back to humans. Heinrich's work [8] and a more recent work [9] reviews speech recognition in the context of human-robot interaction. Speech also provides implicit form of intent communication by understanding emotions [10].

Gestures are visual cues that can be either explicit or implicit. They are effected by the head, arms or body of the operator. Explicit gestures include eye gaze [11] or communicative gestures such as pointing to draw attention [11], [12] or the use of pre-defined signs [13][14]. Intentions can be also derived implicitly while manipulating objects and via facial expressions (gesture) [15] or the head pose [12] of the operator. Sensory communication involving inputs from inertial measurement units (IMUs), encoders also can be a source of explicit intent information, in that they can measure the orientation or position of an end-effector, for example that reveals the intent [16]. As an example, the intention of a robot that was going to pick one of two parts can be revealed from the trajectory it begins to take obtained from its motor encoders.

The use of graphical or visual interfaces to communicate intent for HRI is not uncommon. The advancement of technology of late have witnessed the use of extended reality communication channels to use the shared environment as a canvas to facilitate communication of intent. The 'digital desk' prototyped by Terashima and Sakane [17] allows two levels of interaction between an operator and a robot via a virtual operational panel (VOP) and an interactive image panel (IIP), both projected on a table separate from the worktable of the robot. While the VOP is used to communicate task dependent operations, the IIP streams the robot's workspace by a separate vision system with which the operator is able to convey target object intentions by touching it with his/her hands. However, as the system was used to for only 'guiding and teaching robot tasks' the intent communication worked primarily in one direction from the operator to the robot. Later, Sato and Sakane [18] extended this further by an 'interactive hand pointer' (IHP) that allowed the operator to point directly at the object in the robot's workspace to convey his/her intentions to the robot (in one direction).

A mobile projector solution was presented by Schwerdtfeger et al. [19] where a laser projector was mounted on a helmet to be worn by the operator. The device projected simple 3D aligned augmentations for welding points on the surface of the part the operator interacts with (a car door) with instruction being provided on a separate stationary computer monitor. Although it was not employed in a collaborative environment, it is important to study this contribution from a technology and ergonomic standpoint to draw first impressions on its feasibility in a collaborative environment. The device was later reported as "too heavy and big" for use as a head-mounted device [20].

Andersen et al. [16] present an object-aware projection technique that takes into account the pose and shape of the object by tracking it in real-time. However, the approach prioritizes the intentions of the robot and there is no mechanism for expressing operator intentions. Leutert et. al. [6] propose aspatial augmented reality system using two projectors, one fixed and the other mobile, for visualizing robot drawing programs and its orthographic pose. Although they propose the use of a tracked device as input to the system, there is no evidence of it coming to fruition. Chakraborti et. al. [21] propose an augmented-reality system wherein the robot projects holograms to the workspace that the operator can interact with using a wearable device (HoloLens). A comparison of the works above has been summarized in Table 1.

TABLE I. Comparison of Projection-Based Intent Communication Solutions with This Work

Author(s)	**Projection-based Intent Communication Mechanism**		
	Modality	***Duplexity (Operator-> Robot Robot->Operator)***	***Required Equipment***
Terashima and Sakane [17]	Vision, Touch	Half-Duplex (O->R)	Desk, Projector, Operator's hand
Sakane and Sato [18]	Vision, Gestural	Half-Duplex (O->R)	Camera, Projectors, Operator's Hand
Schwerdtfeger et. al [19]	Vision	-	Camera (Helmet Tracking System), Projector, Head Mounted Helmet
Andersen et.al.[16]	Vision	Half-Duplex (R->O)	Projector, Camera
Leutert et. al [6]	Vision	Half-Duplex (R->O)	Projectors
Chakraborti et. al. [21]	Vision	Full-Duplex (O->R, R->O)	HMD (HoloLens), Operator's Hand
This Work	Vision, Touch, Gestural	Full-Duplex (O->R, R->O)	Projector, Camera, Operator's Hand

From reviewing literature, we identify the need for enabling bi-directional communication of intent between collaborative agents. Augmented-reality solutions using wearables can be unsuitable for industrial environments as they can be non-ergonomic for collaborative assembly [22]. The advantage with conventional reality pure projection-based interfaces is that they are non-obtrusive, and do not negatively impact ergonomics like their augmented-reality counterparts albeit requiring a flat surface for projection with limited field of view. Pure projection-based interfaces, to the best of our knowledge has never been implemented to allow bi-directional communication between humans and robots and this study takes this up as its objective.

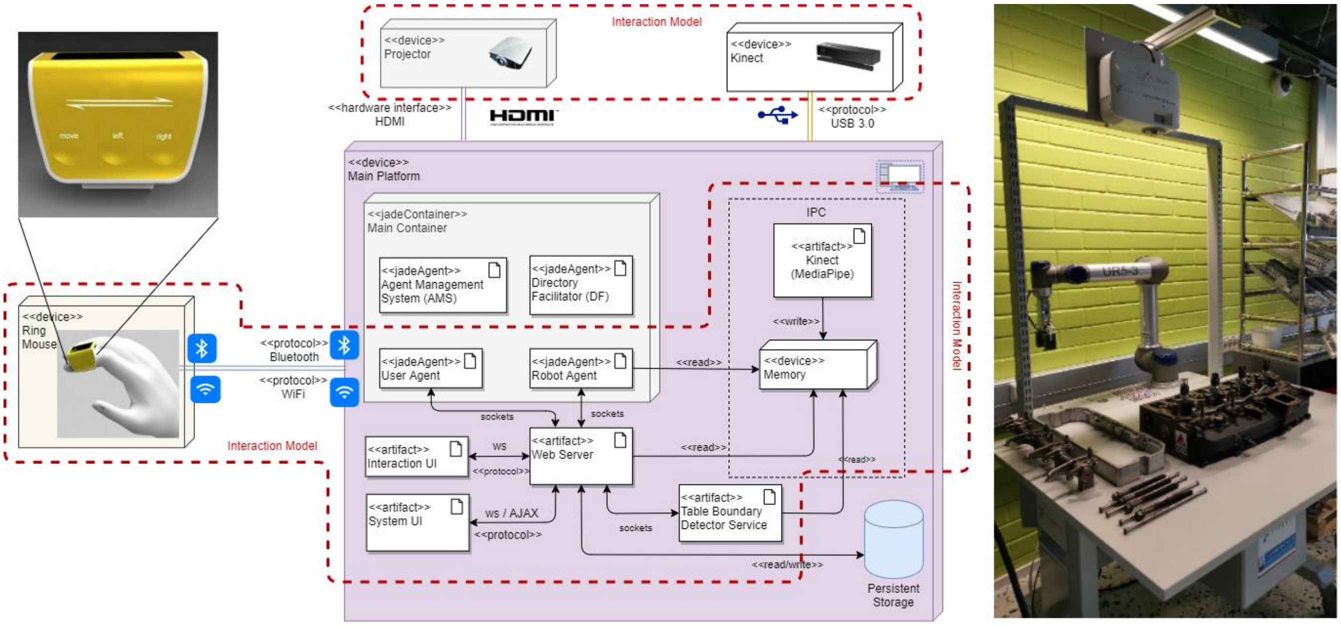

Fig. 1. Deployment Diagram representing the basic architecture including the wearable ring mouse, and the physical setup (right).

III. Research Objectives and Research Setting

To address the gap identified, the objective of this study is:

- to develop a mixed-reality interaction model that facilitates full-duplex human and robot communication.
- to base the communication on prevalent agent-based standards.
- to enable interactions in a minimally invasive manner taking into account the ergonomics and usability.

This paper describes the interaction model and explains its entailed communication from a technology standpoint (*the how*). What exactly is communicated (*the what*) is intentionally left out and is part of an information model that is being currently developed.

The research is carried in a laboratory environment shown in Fig. 1 (right). It consists a DLP projector, and a Kinect Camera (RGB-D) mounted atop a height adjustable table that acts as a collaborative working space between a table-mounted UR5 collaborative robot and a human operator.

IV. Hardware and Software Components

This section presents background information by briefly introducing the technology, hardware and software used in the interaction model.

A. *Java Agent Development (JADE)*

JADE is a software middleware fully developed in JAVA that is used to build distributed multi-agent systems based on a peer-to-peer communication architecture [23]. A *container* is one instance of the JADE run-time inside which a JADE agent resides and provides the necessary services for agents to function (Fig. 1). A special *main container*, is the first container that exists on start-up and hosts two special components; the *Agent Management System* (AMS) that is responsible for the functioning of the *Agent Platform* (not shown) such as creating and deleting agents and a *Directory Facilitator* (DF) that provides yellow pages services to agents in order to discover services provided by other agents. A detailed architecture of JADE is out of the scope of this study but the interested reader is encouraged to refer to the afore cited publication.

B. *Kinect*

The Kinect sensor (v2) is a hardware device that incorporates a RGB Camera and detectors that maps depth through time-of-flight (ToF) of light from a separate IR emitter be-tween a distance of 0.5m to 4.5m. The IR Camera resolution is 512 x 424 pixels while the RGB camera resolution is 1920x 1080 both which operates at 30fps.

C. *Ring Mouse*

Ring mouse is a device worn by the human operator (Fig. 1.) to be able to simulate mouse clicks on the Interaction UI. It is a commercial off-the-shelf hardware that has options to simulate a left mouse click, a right mouse click and scroll (Fig. 1. left-bottom).

D. *MediaPipe*

MediaPipe is an open source project by Google to build perception pipelines as a graph of reusable nodes called *Calculators* [24]. These *Calculators* are connected in a *Graph* by means of data *Streams* that are essentially a time-series of a basic data flow unit known as a *Packet*. The ecosystem of modular calculators and re-usable graphs allows for rapid prototyping by swapping in and out calculators that share common interfaces with different functionalities. A detailed description of the framework is out of scope of this study but the interested reader is encouraged to read the work of Lugaresi et. al. [24].

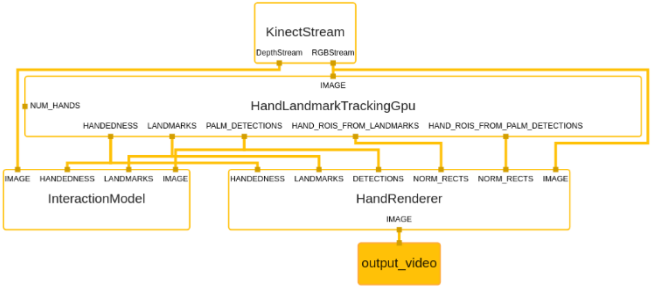

Fig. 2. MediaPipe graph that implements the Interaction Model

V. Interaction Model

A. Architecture

The basic architecture of the interaction model is shown in Fig. 1. It consists of a typical calibrated Kinect camera-projector setup that is mounted atop the worktable. The basic idea is that a purpose built web-app is projected on to the worktable that both agents use to interact with each other (Fig. 6). The web-app is tightly integrated with an agent-oriented middleware, JADE, that allows for a standards-based communication between agents. The architecture is part of a larger framework [25] that uses a knowledge based engineering software to realize a digital thread for a product aware human robot collaboration [26].

B. Input

The input to the interaction model is obtained via the RGB sensor of the Kinect that monitors the environment including the worktable that the user interface is projected upon. A hand detection algorithm implemented using MediaPipe [24][27] looks for the human operators hands at run-time. The coordinates of the detected hands in the image plane of the RGB sensor is transformed to the Projector plane by a pre-computed camera-projector homography [28]. A planar homography is a bijective mapping (having one-to-one correspondence) between projective spaces of a point lying on a plane in the vector space from which the projective space is derived. In other words, it maps the pixel coordinates from the image plane of the RGB sensor of the camera to the pixel coordinates of the projector's image plane with respect to the plane it is projected onto(the table).

Fig. 2 shows the MediaPipe graph used to implement the interaction model as rendered by the MediaPipe Visualizer. The *HandLandMarkTrackingGpu* and the *HandRendererNode* is reused from the framework while *KinectStream* and the *InteractionModel* nodes are implemented as part of this research work. The *HandLandMarkTrackingGpu* node detects the operators hand while the *HandRenderer* node just outputs a rendering of the hand juxtaposed with the detected landmarks if any and is something the interaction model can function without to fulfil the functions it is currently designed for. However, we may use this in future to record interactions for viewing by admin personnel or for upstream use for the designer. An output frame rendered by *theHandRenderer* node is shown in Fig. 3. Further details pertaining to the working of these two nodes is considered out of scope and will not be discussed any further. The interested reader is encouraged to read the work of Zhang et. al [27].

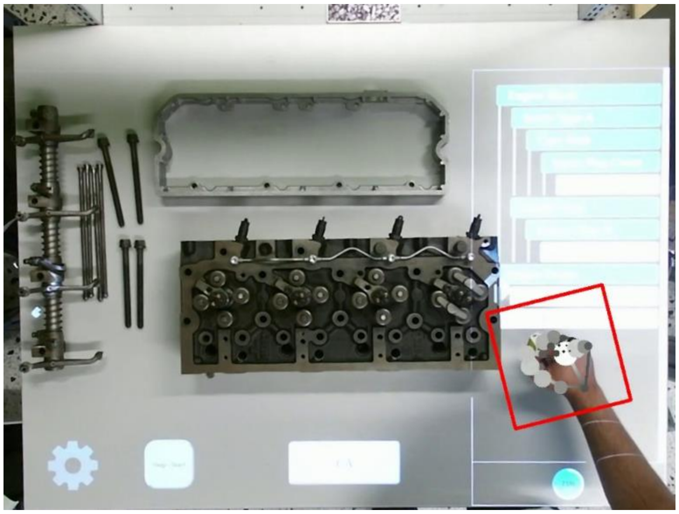

Fig. 3. User Interface as viewed from the Kinect Sensor (*HandRenderer* Node output).

The *KinectStream* node uses the open source *libfreenect2* library to obtain the raw RGB and depth frames from the Kinect Sensor. The *InteractionModel* node uses this depth stream and the landmarks from the *HandLandMarkTrackingGpu* to implement the core functionality of the *InteractionModel* node. The depth of the operator's hand is first checked against a pre-defined threshold (based on the height of the worktable) which, if considered valid, writes the hand landmarks to a location memory that is shared with the Web Server. The webserver then maps the hand landmarks to mouse movements on the main platform that hosts the user interface. While the hand detection rules out the possibility of unintentional interactions by any object, the height threshold prevents the operator from accidentally interacting with user interface while manipulating the product. Accidental interactions are further ruled out by a small wearable ring mouse that is worn by the human operator where the operator has to effect a physical mouse left-click by touch to interact with the elements of the user-interface when used in the mouse mode. The ring mouse model that we currently use (Fig. 1, left-top) has touch buttons for this purpose. This is a necessity that stems from the use of a web-browser as an interface as there is no going around mouse events to interact with web-browser elements. However, the user also has the option to use gestures for interactions. A transition from an open face of the operator's palm to a closed fist would simulate a mouse left-click while the transition vice-versa simulate its release. However, at any point in time only one of the two modes would remain active, and the operator will be expected to choose between them. Thus, the interaction model could support a full-duplex communication model without the need for any input device. The *InteractionModel* node communicates the hand land-mark location, the RGB image and the handedness to other interested software artefacts via memory it shares with them, a form of inter-process communication. For example, a robot that needs perception of the environment could read the RGB stream from the shared memory.

C. User Interface

This section reports the latest working prototype of the user-interface and is likely to change in subsequent iterations to

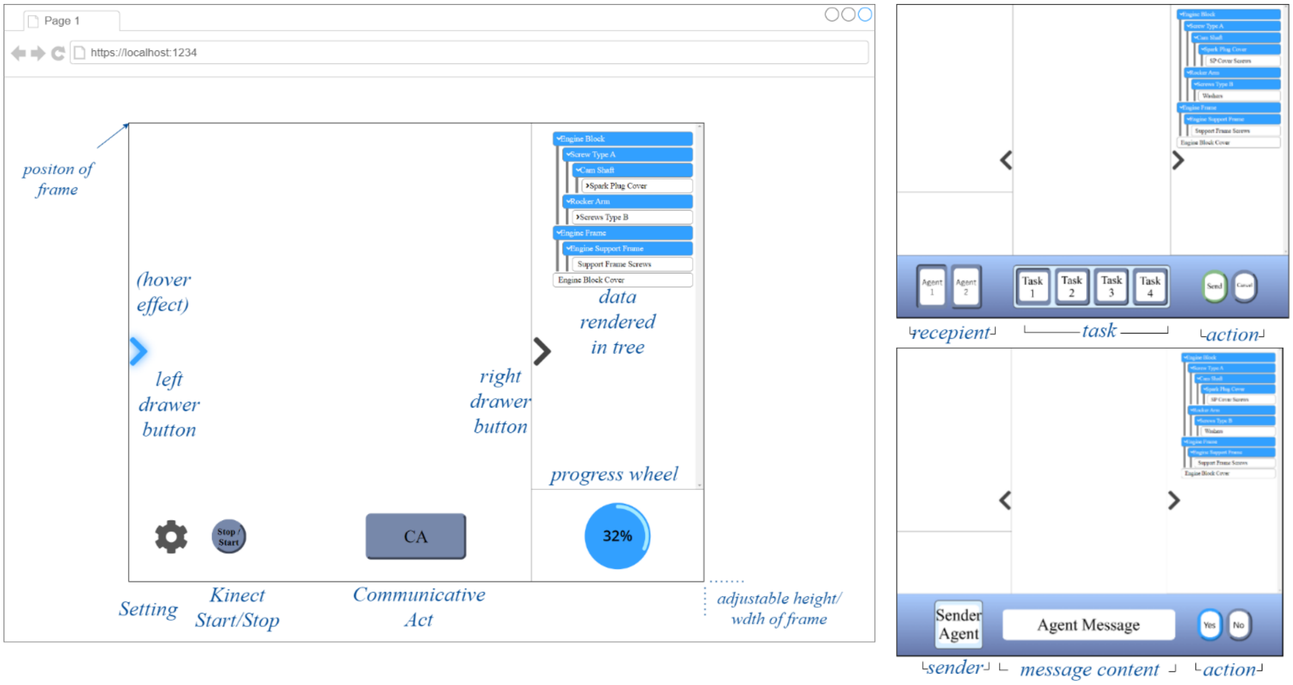

Fig. 4. Initial Screen (left), Screen after the CA button is pressed (right, top), Screen with an interaction message (right, bottom).

incorporate more semantics and functionality. The user interface is a web-browser that is projected onto the worktable. The front-end development is purpose-built and is kept simple and uses only vanilla JavaScript, HTML and CSS. The elements of the user-interface presented herein act merely as placeholders and used for illustrative purposes to explain the functionality only and not intended to be readable.

The operator is first presented with a screen that asks to be made in full screen (not shown). Full screen is an implicit requirement that stems from the computed Camera-Projector Homography for the interaction model. Fig. 4 (left) shows the screen that the operator is presented with once it is made full screen (the browser bar - menu bar, address bar, tool bar, shown in Fig. 4 (left) is shown only for context to visualize a web browser). Further, the background in reality is a black HTML5 canvas for high contrast with white text to aid readability. In Fig. 4, the background is shown as white to be publication friendly. Choosing black as colour of the canvas also means that the projector does not illuminate the operator's hand which is necessary as any other color would affect the performance of the MediaPipe model that detects the operator's hand. The part of the UI that is projected onto the worktable lies completely within a single *div* HTML element as shown. As such it has the ability to positioned by the top left coordinate and also adjusted for its width and height. This allows the UI to be projected to worktables at different positions and of different sizes. All elements have a blue hover effect for visual feedback for the operator that gets activated at an invisible radius around the element. The need for activating visual elements beyond their visual boundaries comes from the fact that input to the interaction model is by tracking the operator's hand that spans over many pixels as opposed to a mouse pointer that can essentially point to a single pixel. This would allow the operator to easily interact with small visual elements and also prevent strict requirements on the location of operator hands to initiate interactions with other visual elements. Also, if the cursor is made invisible, the user can rely only on the hover effects alone as a visual feedback to interact with the user-interface.

The UI has two resizable side drawers on both sides that can be drawn open by arrow buttons. The right drawer visualizes any information in a tree structure and has a progress wheel at its bottom. This is envisioned to present the task sequence and its completion percentage respectively. Currently, there are three other buttons, a settings button to manage settings, a button to start or stop the Kinect sensor and a button to initiate a communicative act (CA). Fig. 4 (right top) shows a screen after the CA button is pressed. It brings up a communication panel that allows to interact with an agent. The agent the operator wants to send the message to is on the left, the current possible or available tasks are at the center and the send or cancel buttons are on the right. Fig. 4 (right bottom) shows a message sent by an agent to the operator. It contains the sender information on the right, the message content in the center and the action on the left. It has the highest priority and is layered above all the other visual elements in the user interface. This is because if an agent wants to communicate to the operator, it should be presented to the operator irrespective of what the operator would be doing.

D. Agent Interaction Process

The process that entails the interactions made possible by the interaction model is discussed in this sub-section. The user interface introduced in the previous sub-section is tightly coupled to an agent-oriented framework JADE that facilitates the communications under the hood. Thus a bidirectional communication needs to be in place between the user-interface and the agents. As mentioned before, what exactly is communicated is intentionally left out and will be reported in a future study detailing the information model . This section

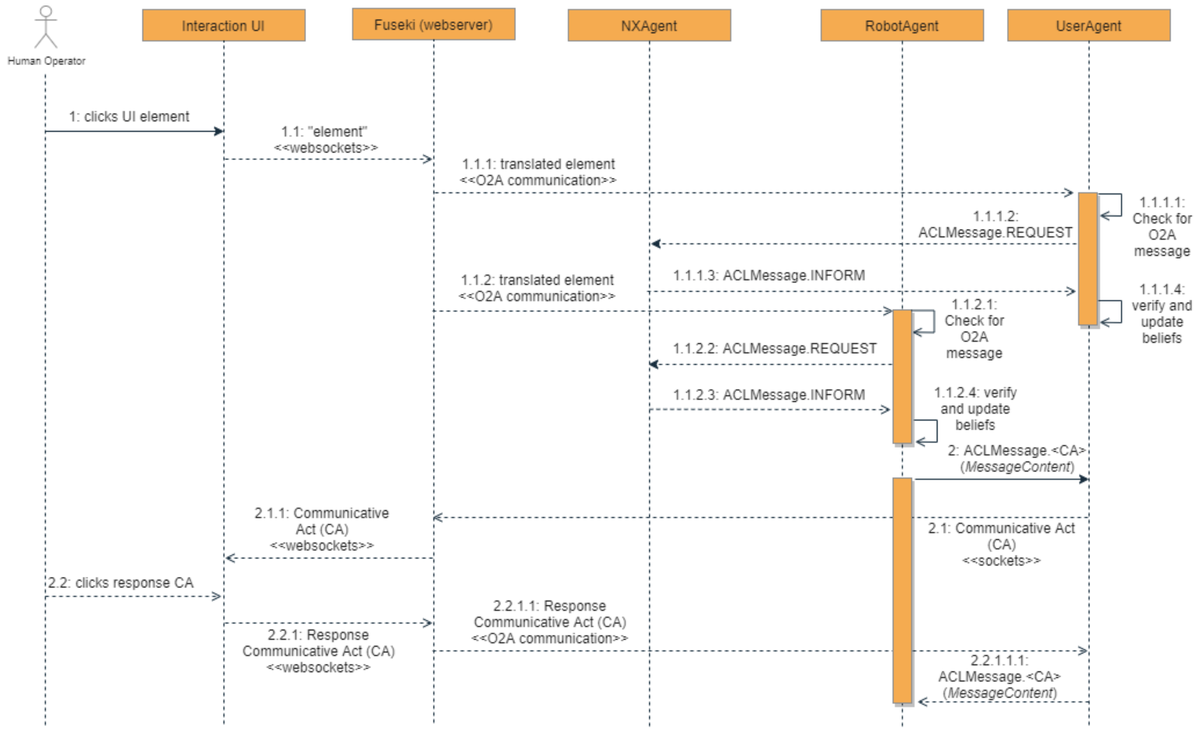
Human Operator
Interaction UI
Fuseki (webserver)
NXAgent
RobotAgent
UserAgent
1: clicks UI element
1.1: "element" <<websockets>>
1.1.1: translated element <<O2A communication>>
1.1.1.1: Check for O2A message
1.1.1.2: ACLMessage.REQUEST
1.1.2: translated element <<O2A communication>>
1.1.1.3: ACLMessage.INFORM
1.1.1.4: verify and update beliefs
1.1.2.1: Check for O2A message
1.1.2.2: ACLMessage.REQUEST
1.1.2.3: ACLMessage.INFORM
1.1.2.4: verify and update beliefs
2: ACLMessage.<CA> (MessageContent)
2.1.1: Communicative Act (CA) <<websockets>>
2.1: Communicative Act (CA) <<sockets>>
2.2: clicks response CA
2.2.1.1: Response Communicative Act (CA) <<O2A communication>>
2.2.1: Response Communicative Act (CA) <<websockets>>
2.2.1.1.1: ACLMessage.<CA> (MessageContent)

Fig. 5. Sequence Diagram the represents the Agent Interaction Process

reports on how such an interaction takes place in the developed interaction model.

1) *UI-Agent Communication*

The agent implements a cyclic behaviour `CheckO2AMessages()` that checks for these objects from the user-interface. A typical such agent interaction process via the interaction model is presented in Fig. 5 where the user initiates the interaction which causes the human and robot agents to fetch some information from the *NXAgent* (design software agent). The process starts with the human operator pairing or connecting the wearable ring mouse with the main platform via bluetooth or Wi-Fi (not shown in figure).

1: Once the operator is wearing the peripheral, the operator clicks a button on the user interface.

1.1: This causes the front-end (Interaction UI) JavaScript to send an appropriate message to the backend webserver via websockets.

1.1.1 and 1.1.2: The server then translates these messages as objects and posts them to the FIFO *O2A* queue maintained by the concerned agents.

1.1.1.1 and 1.1.2.1: Each agent as mentioned before, implements a cyclic behaviour that constantly checks for the *O2A* communication. Each agent has coded logic that allows them to recognize the kind of Java objects in their *O2A* queue. In this example case in Fig. 5, it is a *communication object* that tells the agent that it needs to communicate to another agent. Thus, it will have fields like *communicative act*, *recipient* and *message content*. Once the agent receives such a java object from its O2A queue, it does exactly that, i.e., initiates a communication.

2) *Agent-UI communication:*

The Agent-UI communication takes place via sockets. The implementation of agents is such that it initiates a websocket connection with the webserver upon creation in the `setup()` method. This bi-directional connection lasts for the entire lifecycle of the agent. It is also useful for sending streaming data such as the joint values of the robot agent. The sequence diagram in Fig. 5 shows such an example of robot initiating a *communicative act* with the human operator to which he/she responds.

2: The robot agent sends a JADE `ACLMessage` to the JADE *User Agent*.

2.1 and 2.1.1: The *User Agent* implements a cyclic behaviour that sends any messages it receives via the socket connection to the *Web Server* that forwards it to the *Interaction UI* interface. The *Interaction UI* handles this incoming connection and displays it for the human operator to respond.

2.2-2.2.1.1.1: The human operator then clicks his/her response on the user interface which then reaches the *RobotAgent* via the UI-Agent communication described previously.

The interaction model thus facilitates a bi-directional agent-oriented communication between the participating agents.

VI. Discussion

The approach taken in this study for communication of intent between agents involves using a web-based user interface for projecting and communicating intentions. Several choices affected the design of the interaction model which are described in this section. The interaction model is part of a larger framework that integrates the product design environment that is the subject of discussion in our earlier work [25]. In this paper, only the interaction model is discussed.

Several quality requirements were addressed. First, the system is modular for reuse for another setup with a different camera input or a trained model. All that needs to be done, is to rewrite the respective nodes in MediaPipe (Fig. 2). MediaPipe allows to swap in/out nodes, provided the input-output relationships of the nodes it interacts with is maintained. MediaPipe is also open-source and cross-platform [24].

The *InteractionModel* MediaPipe node shares data via shared memory that allows access by any process running on the same computer. This in theory means that it is more performant than middlewares such as ROS, has no dependencies and also scales easily to additional agents if needed. The only problem with shared memory is that they shared the same physical memory so they should be connected to the same machine. However, this should not be an issue as the physically collaborating agents would in be in close proximity anyway.

The choice of a web-browser was a conscious one. Most systems come with browsers built-in and thus has no dependencies from the GUI point of view and is also interoperable as the languages it uses, vanilla JavaScript, HTML and CSS, have been standardized, works cross-platform and undergone a couple of decades of improvement. Such improvements include the building of responsive interfaces owing to the proliferation of mobile devices. Using responsive web design principles means that it is possible to project to worktables of different dimensions (caveat - of only rectangular or square shapes) and thus be able to maintain a reduced cognitive load on the operator without any change in configurations. Further, web-browsers enable to enrich the operator's experience by using a wide range of rich 3rd party libraries. A specific one that we intend to use in an upcoming information model that we are currently developing is three.js3, that allows to render 3D STL files with the product and manufacturing information (PMI). This in principle would create better situational awareness for the operator and contribute to the overall efficiency of the manufacturing process. Using a browser also reduces the development time as it reduces the design of the user-interface to a front-end web development problem, a problem that in today's day and age has solution architects aplenty.

The integration of JADE in the interaction model has its advantages too. JADE provides a set of primitives grounded on speech act theory known as *performatives* or *communicative acts* which is essentially a classification of a message based on the implied action. Examples of this include *inform*, *request*, *agree*, *refuse*, etc. Each of these are among 22 communicative acts described by the FIPA Communicative Act Library Specification [29] and have a formal semantics based on modal logic that can manipulate the mental model of the sender and the receiver agents. Such manipulations can be used trigger inference models that enables agents to exhibit intelligent behaviour. What this also allows for is the possibility to specify a predefined sequence of messages that entails a specific interaction what is known as an interaction protocol. An operator can use these well-defined protocols to communicate to carry out collaborative tasks without having to guess or understand implicitly from the behaviour of the robot. Let us consider the case of a collaborative assembly task as an example. A robot agent that wants to collaborate with an operator would initiate a request using the FIPA Propose-Protocol [30] using the *propose* communicative act, to which the operator would respond with an *accept-proposal* or *reject-proposal* communicative act. Meanwhile, the robot or human agent at any time could decide to cancel its intention to collaborate using the FIPA Cancel Meta-protocol in a manner acceptable to both. Thus, the interaction model allows for an explicit and standardized means of communication of intent via these interaction protocols. Leveraging such explicit communication channels understandable to humans potentially in conjunction with other modalities would in theory help reduce the ambiguities in intent recognition. Further, the benefits of using standardized interaction protocols and JADE in general means that the existence of the agents may not beknown beforehand and can be discovered at run-time, can be developed independently while being guaranteed to work together.

VII. Conclusion

The work presented in this paper presents a novel approach towards enabling standards-based explicit bi-directional intent communication between collaborative human and robot agents while embracing open-source initiatives. It takes the functionality of pure projector based interface a step further and provides a non-obtrusive and effective alternative, especially in structured use-cases such as that of a collaborative product assembly. Such channels of communication maybe used in addition to other modalities to realize multi-modal means of intent communication. We identify room for improvement in the current iteration of the interaction model reported in this study. There needs to be a mechanism to project the user-interface within the boundaries of the worktable in the projector's field of view. Currently, although the UI (div) can be positioned anywhere and adjusted for height and width, it is hard-coded to align atop the worktable. Work is underway to develop machine-vision based algorithm for it to be done automatically (Table Boundary Service in Fig. 1 (left)).

While this work has established mechanisms of communication, i.e., the interaction model, the information model that defines the semantics of communication remains part of future work. The information model includes architecture of the agent's mental attitudes, how the agents converge upon shared mental representations of knowledge dynamically at runtime, the vocabulary of communication, ontology, etc. It would enable agents to adapt to dynamic role changing with respect to the task in context based on their skills. The information model also includes the visual semantics of the user interface. For example, notifying message that an operator may choose to passively ignore would differ from a request to collaborate visually by colour. As another example, critical errors or warnings maybe represented in another colour (red perhaps) that draws immediate attention. The current work presents the barebones of the UI and such semantics remain as future work as part of the information model.

References

[1] B. Hayes and B. Scassellati, “Challenges in Shared-Environment Human-Robot Collaboration,” 2013.

[2] L. Wang *et al.*, “Symbiotic human-robot collaborative assembly,” *CIRP Ann.*, vol. 68, no. 2, pp. 701–726, 2019.

[3] N. Nikolakis, V. Maratos, and S. Makris, “A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace,” *Robot. Comput. Integr. Manuf.*, vol. 56, pp. 233–243, 2019.

[4] Z. Kemény, J. Váncza, L. Wang, and X. V. Wang, “Human--robot collaboration in manufacturing: a multi-agent view,” in *Advanced Human-Robot Collaboration in Manufacturing*, Springer, 2021, pp. 3–41.

[5] A. Watanabe, T. Ikeda, Y. Morales, K. Shinozawa, T. Miyashita, and N. Hagita, “Communicating robotic navigational intentions,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5763–5769.

[6] F. Leutert, C. Herrmann, and K. Schilling, “A spatial augmented reality system for intuitive display of robotic data,” *ACM/IEEE Int. Conf. Human-Robot Interact.*, pp. 179–180, 2013.

[7] A. E. Uva, M. Gattullo, V. M. Manghisi, D. Spagnulo, G. L. Cascella, and M. Fiorentino, “Evaluating the effectiveness of spatial augmented reality in smart manufacturing: a solution for manual working stations,” *Int. J. Adv. Manuf. Technol.*, vol. 94, no. 1, pp. 509–521, 2018.

[8] S. Heinrich and S. Wermter, “Towards robust speech recognition for human-robot interaction,” in *IROS2011 Workshop on Cognitive Neuroscience Robotics (CNR) 2011*, pp. 29–34.

[9] C. Deuerlein, M. Langer, J. Seßner, P. Heß, and J. Franke, “Human-robot-interaction using cloud-based speech recognition systems,” *Procedia CIRP*, vol. 97, pp. 130–135, 2021.

[10] J. G. Razuri, D. Sundgren, R. Rahmani, A. Larsson, A. M. Cardenas, and I. Bonet, “Speech emotion recognition in emotional feedback for Human-Robot Interaction,” *Int. J. Adv. Res. Artif. Intell.*, vol. 4, no. 2, 2015.

[11] C. Rich, B. Ponsleur, A. Holroyd, and C. L. Sidner, “Recognizing engagement in human-robot interaction,” *5th ACM/IEEE Int. Conf. Human-Robot Interact. HRI 2010*, pp. 375–382, 2010.

[12] R. Stiefelhagen *et al.*, “Enabling Multimodal Human–Robot Interaction for the Karlsruhe Humanoid Robot,” *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 840–851, 2007.

[13] D. Ryumin, D. Ivanko, A. Axyonov, I. Kagirov, A. Karpov, and M. Zelezny, “Human-Robot Interaction with Smart Shopping Trolley Using Sign Language: Data Collection,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2019, pp. 949–954.

[14] S. Singh, A. Jain, and D. Kumar, “Recognizing and interpreting sign language gesture for human robot interaction,” *Int. J. Comput. Appl.*, vol. 52, no. 11, 2012.

[15] Z. Liu *et al.*, “A facial expression emotion recognition based human-robot interaction system,” *IEEE/CAA J. Autom. Sin.*, vol. 4, no. 4, pp. 668–676, 2017.

[16] R. S. Andersen, O. Madsen, T. B. Moeslund, and H. Ben Amor, “Projecting robot intentions into human environments,” in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2016, pp. 294–301.

[17] M. Terashima and S. Sakane, “A Human-Robot Interface Using an Extended Digital Desk,” no. May, pp. 2874–2880, 1999.

[18] S. Sato and S. Sakane, “Human-robot interface using an interactive hand pointer that projects a mark in the real work space,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 1, no. April, pp. 589–595, 2000.

[19] B. Schwerdtfeger and G. Klinker, “Hybrid Information Presentation : Combining a Portable Augmented Reality Laser Projector and a Conventional Computer Display,” 2007.

[20] B. Schwerdtfeger, D. Pustka, A. Hofhauser, and G. Klinker, “Using laser projectors for Augmented Reality,” *Proc. ACM Symp. Virtual Real. Softw. Technol. VRST*, pp. 134–137, 2008.

[21] T. Chakraborti, S. Sreedharan, A. Kulkarni, and S. Kambhampati, “Projection-Aware Task Planning and Execution for Human-in-the-Loop Operation of Robots in a Mixed-Reality Workspace,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 4476–4482, 2018.

[22] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, and J.-K. Kämäräinen, “AR-based interaction for human-robot collaborative manufacturing,” *Robot. Comput. Integr. Manuf.*, vol. 63, p. 101891, 2020.

[23] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. .

[24] C. Lugaresi *et al.*, “MediaPipe: A Framework for Building Perception Pipelines.”

[25] J. David, A. Lobov, E. Järvenpää, and M. Lanz, “Enabling the Digital Thread for Product Aware Human and Robot Collaboration - An Agent-oriented System Architecture,” in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021. pp. 1011-1016

[26] J. David, E. Järvenpää, and A. Lobov, “Digital Threads via Knowledge-Based Engineering Systems,” in *2021 30th Conference of Open Innovations Association FRUCT*, 2021, pp. 42–51.

[27] F. Zhang *et al.*, “MediaPipe Hands: On-device Real-time Hand Tracking.” 2020.

[28] G. Falcao, N. Hurtos, and J. Massich, “Plane-based calibration of a projector-camera system Plane-based calibration of a projector-camera system,” no. February, 2015.

[29] “FIPA Communicative Act Library Specification.” [Online]. Available: http://www.fipa.org/specs/fipa00037/SC00037J.html#_Toc26729714. [Accessed: 30-Jul-2021].

[30] “FIPA Propose Interaction Protocol Specification.” [Online]. Available: http://www.fipa.org/specs/fipa00036/SC00036H.html. [Accessed: 30-Jul-2021].