DOI: 10.3217/978-3-85125-663-5-02

Intuitive Human Machine Interaction based on Multimodal Command Fusion and Interpretation

Leopold Hiesmair[1], Matthias Plasch[1], Helmut Nöhmayer[1] and Andreas Pichler[1]

Abstract— **The drastic transition from mass production to mass customization and small lot-sizes in production industry, requires intuitive interaction, programming and setup approaches for machinery and robotics in order to reduce setting-up time or adaption effort. Multimodal data fusion and analysis is considered as a potential enabling technology to achieve intuitive human machine interaction. Our work focuses on robust interpretation of commands, issued by a human actor, which are combined of single attributes created from different multimodal channels. The presented approach is demonstrated using an example of human robot interaction, where the user interacts with the robot to setup a robotic process sequence.**

I. Introduction

Despite the increasingly simple programming interfaces that are available for production machinery, a deeper understanding of these systems is needed. For the programming, maintenance and adaption of processes, a highly qualified technician is required in many cases, who is only available to a limited extent. Therefore, the setting-up time is increased, which consequently reduces the flexibility. Our goal is to enable an intuitive communication with various modalities so that non-expert users are able to interact with and to program production machineries. This paper presents the conception and development of a multimodal data recognition and fusion system, to interpret and process commands issued trough an human actor. The multiple command parts provided by various channels, such as speech, gesture and haptic interaction, are analyzed and interpreted in order to generate a valid command statement. This fusion system is intended to be used for human machine interaction in general. Within the scope of this work, an example was developed which considers the domain of human robot interaction (HRI). The resulting *Command Fusion and Interpretation* (CFI) module enables robust control of a robot system based on simple interaction commands. The outline of the paper is as follows. Section II gives an overview on the state of the art in multimodal data analysis and discusses related work. In section III, requirements to multimodal data fusion and command interpretation are presented, following the description of our implementation approach in sections IV and V. Section VI presents the testing environment test procedures for our approach. The paper is concluded by a brief discussion about the results and explaining future work items in section VII and VIII.

II. State of the art and related work

Multimodal data fusion and processing has gained a lot of interest in different research fields, especially in the areas of multimedia analysis [1][9][15][23][12], human machine (HMI) and human robot interaction (HRI) [22][3][10][8][6]. Atrey et al. [4] provide an extensive survey on fusion strategies for multimedia analysis, discuss general characteristics and common challenges arising during the implementation of multimodal data fusion.

According to [4] and [19] data fusion is performed on either *feature level (early fusion)* or *decision level (late fusion)*. *Feature level fusion* is often applied in case of strongly coupled inputs (like e.g. voice and lips movement). The extracted information of each modality is combined into one single vector, which is sent to the interpretation unit [23]. *Decision level fusion* combines local feature decisions into a vector. To derive a final decision, a synchronization between the different feature decisions is required, due to the different latencies of the classifiers [16].

The fusion approaches can be categorized in (application references are mentioned in brackets): a) *Rule based* ([11][5]) b) *Classification based* ([1][9][12]) and c) *Estimation based* ([15]). A detailed description of these categories is provided by Atrey et al. [4].

Rule based fusion approaches can be realized based on Definite Clause Grammars (DCGs) [17]. DCGs are proven to be helpful to describe natural language and are built-in features of first-order logic languages like Prolog [7]. A comprehensive overview on grammars for multimodal data processing is provided in [7]. Shimazu et. al. [21] proposed MultiModal DCGs (MM-DCGs), which provide means to express chronological constraints and handle multiple modalities. In [7] an approach for generating multimodal grammars is presented, which is able to add additional semantics to grammar definitions.

HRI is realized in different forms, depending on the information that has to be communicated, and dependent on the role of the human (e.g. supervisor or operator) [20]. The information exchange is considered as a main topic of HRI research, to enable intuitive and user-friendly interaction. Similar to [13] we consider HRI based on multiple modalities as relevant, to realize a more complex information flow. This is needed in situations where parameters (like speed) should be communicated at the same time as higher level commands, like a coarse moving direction. In such cases using a single modality is not enough to express the intention. Within the HRI domain, Sucar et al. [22] applied fusion of speech

[1] Profactor GmbH; Im Stadtgut A2, 4407 Steyr-Gleink, Austria; Email: name.surname@profactor.at

and gesture data to generate commands valid to control motion of a robot, based on intuitive motion instructions. In [3] and [2] *incremental fusion* approaches for HRI are targeted, where incremental means that distinct multimodal data packages are processed as they are being received. This also requires an incremental generation of command hypotheses until a final statement can be found. Ameri et. al. [3] stress the necessity of weighting the different multimodal channels to cope with error prone modalities. The authors in [18] focus on a bidirectional interaction approach (pointing gestures, voice, status display), to enable confirmation of the interpreted command to achieve high accuracy.

The works [19] and [14] introduce general architectures to realize robust multimodal HRI. Rossi et. al. [19] implemented a decision level and classification based approach using a Support Vector Machine within the fusion layer. Support for an arbitrary number of modalities is given. The work in [14] focuses on deep learning based feature analyses units, to classify data of multimodal channels (body posture, hand motion, voice commands). A late fusion engine is part of future work topics. Multimodal interaction with a group of robots is targeted in [6]. Decision level fusion is applied based on Naive Bayes classifiers. This approach requires a three-step training process: a) one step to train the unimodal classifiers, b) one step for the command recognition system (structures of the possible commands) and c) one step to adapt the thresholds for the command hypotheses.

Our work targets the development of a multimodal data fusion and interpretation architecture based on DCGs in order to generate valid commands for human machine interaction. In this paper we consider the application of controlling a robot based on commands issued by a human through multimodal channels. Those commands are based on keywords that are intuitive to humans (e.g. move up, slower, stop, and others), similarly as proposed by [22]. Although DCGs provide truly less expressiveness like explained in [7][21], we argue that DCGs are still sufficient to perform multimodal data fusion with reasonable results, for the reasons as follows.

- Available commands including their structure and parameters are known in prior for the automation or robotics domain. There is no requirement for generating the validation grammar or to train command classifiers. Determined grammars are easier to understand and lead to determined results.
- Temporal constraints for multimodal fusion can be considered using timeouts, alternatively to applying MM-DCGs. As data fusion and hypotheses are performed continuously during sensing, timeouts basically specify the time duration for a full command to be issued.

The presented solution also allows for online adaptation of command structures during runtime, thus providing a high grade of flexibility. Our work focuses on multimodal data fusion and interpretation of a valid command. Classification of the data emitted through multimodal channels is not in the scope of this work.

III. Requirements

In order to realize robust command interpretation based on multimodal inputs, requirements as follows were defined for the implementation of the CFI module.

- **Partly reception of commands**
  A command (e.g. `move object1 to location2`) can be partly received (command parts, henceforth *attributes*) in a random order and from different input channels.
- **Input validation**
  All input streams need to be checked for plausibility using a collection of rules. Invalid attributes need to be ignored.
- **Confidence level for unreliable sources**
  Most recognition systems indicate a probability measure in relation to the understood attribute. The CFI module has to analyze this confidence level and decide whether it is sufficient or not. The threshold depends on the type of information and the assigned source.
- **Prioritization of commands and channels**
  In the case of multiple valid command hypotheses, operations with higher safety level have to be prioritized. Futhermore, if identical attributes are received from different multimodal channels, but not corresponding to each other (e.g. `move up down`), the system has to prioritize the channel with the higher quality.
- **Flexibility through adjustable parameters**
  Each configuration parameter (e.g. priorities, confidence levels, number of input channels, and others) has to be adjustable during the execution.
- **Feedback and none-feedback mode**
  A feedback mode to present the resulting command to the user is required. The user has the opportunity of aborting the command execution.
- **Heart Beat signals**
  Motion commands are mostly incrementally, i.e. that the system executes a command as long as it is active. A Heart Beat signal maintains the active state.
- **Adaptable command definition during Runtime**
  In different machine operating states (e.g. automatic, hand move, and others), specific commands are allowed. Therefore, a functionality to enable or disable commands during runtime is necessary.

IV. Approach

The general approach targeted by the CFI module is to analyze the received attributes in an incremental fashion. Using such a strategy, a robust way of fusing multimodal data inputs, and generating a valid command statement is achieved. Based on the defined requirements, the approach as described below was realized.

A. Architecture

Figure 1 illustrates a scheme of the architecture of the CFI module. On the left side, different input channels are listed. The blocks on the top of the figure represent the configurability and feedback functionality. A simplified description of the

CFI module's processing sequence is depicted at the bottom. The core functionalities are presented as block diagram in the image center. The meaning of the functional units will be described in section IV-C. The functional workflow of the CFI module is as follows:

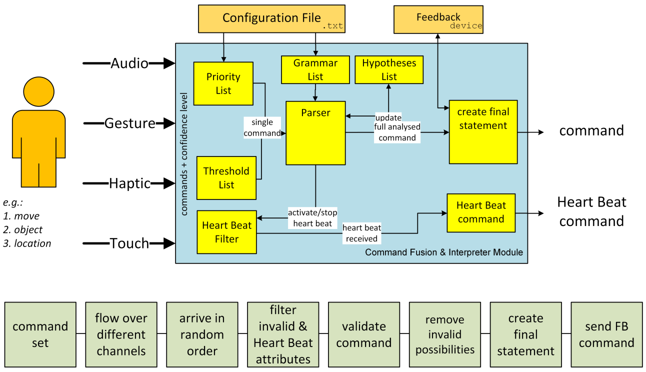

Fig. 1. Architecture of the CFI module

1) The user describes the desired command using different modality channels which are captured by the recognition systems.
2) All created command attributes are transmitted over several input channels to the CFI module.
3) The arrival times of the attributes are random and can overlap.
4) The CFI module filters invalid and active Heart Beat attributes to ensure a correct interpretation of the commands.
5) All attributes are validated by using the *Grammar List* and the *Threshold List*.
6) Each incremental step of validating command structures, the *Hypotheses List* is repetitively updated by removing invalid hypotheses.
7) If a command is fully received and analyzed, the module creates a final statement.
8) The created final statement is sent to the execution module.

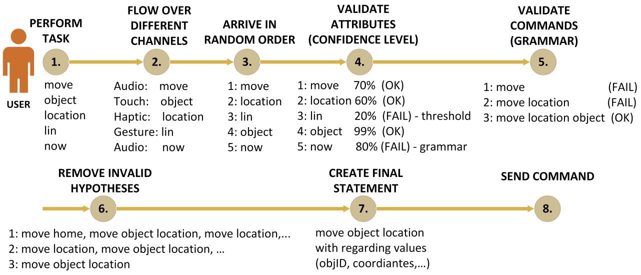

Fig. 2. Command fusion example

B. Working Principle

In order to get a better understanding of the CFI module, the following example in Figure 2 is considered, to explain the individual steps. The steps 1 to 3 are covered by individual recognition systems for multimodal channels and can not be influenced by the CFI module. Each command attribute arrives at the module in an random order and will be immediately validated in step 4. In this example, the invalid attribute `now` (not defined in the Grammar List) and the unreliable attribute `lin` (confidence level below threshold) are ignored. The incremental validation of the command is taken place in step 5 and 6, where each validation in step 5 creates command hypotheses in step 6. At every iteration, an attribute is appended to the command structure and hypotheses are generated. After all attributes are appended and validated, only one hypothesis is left and the final statement can be created and sent to the execution unit, as shown in step 7 and 8.

As mentioned in section III, Heart Beat signals are used to maintain the active state of a command (e.g `move up`). Figure 3 illustrates an example of a Heart Beat usage. Using the *Heart Beat start attribute* `up` repetitively, the active state is maintained. The defined interval time represents the maximum time span between two appearing Heart Beat attributes. If this time span is exceeded, the command will be automatically stopped, as highlighted with the expressions $T1 + 4s$ or $T3 + 4s$. Additionally, the command can be stopped immediately by using the *Heart Beat Stop attribute* `finish`, as shown in the last example of Figure 3.

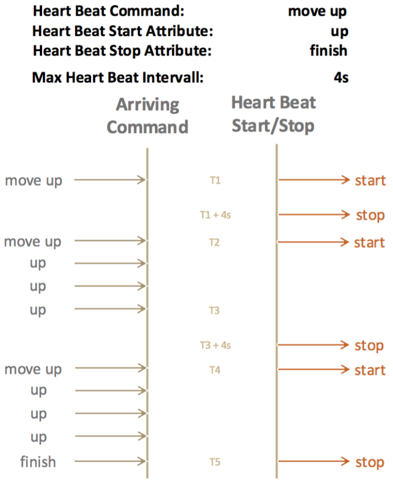

Fig. 3. Usage of Heart Beat attributes (e.g. start/stop)

C. Submodules and Functionalities

The depicted modules in Figure 1 basically consist of different lists and components that interact with each other. The purpose of each block/list is described below:

- **Input Channels**
  The input channels consist of different modalities. Those can be divided into unreliable channels, where recognition is based on trained classifiers (e.g. speech, gesture, haptic, etc.), and reliable channels with defined states (e.g. touch panel, button, etc.).
- **Priority List**
  The Priority List defines every available command with

their respective priority level. This level is used for prioritizing attributes in case of simultaneously arriving commands.

- **Grammar List**
  Unreliable modalities can cause unintended commands and attributes. Therefore, all invalid commands and attributes are filtered in order to prevent invalid states. The Grammar List summarizes all valid command structures with the valid combinations of attributes describing a possible command (e.g. `move object location`) using DCGs. Additionally, this list defines the output structure of the final statement.
- **Hypotheses List**
  After every validation of a command structure, a pool of hypotheses is created and stored into the Hypotheses List in order to reflect all further possibilities. The updated Hypotheses List indicates whether a command is fully received or not. After updating this list three cases are possible, depending on the list size:
  - Empty: command structure is not valid
  - Single entry: command structure is fully received
  - Multiple entries: command structure can be extended with further attributes
- **Threshold List**
  Several recognition systems use a distribution for the likeliness of the understood inputs (commands or attributes). These inputs are received with a corresponding confidence level and will only be accepted over a certain threshold. By analyzing the Hypotheses List a certain number of command attributes can be predicted to adapt the threshold levels for the respective attributes.
- **Parser**
  The multimodal Parser plays a central role in terms of analyzing the gathered information. It uses all lists for validating the commands and generating hypotheses. The algorithm is described in section V-B.
- **Output Sentence Structure**
  If a command is fully received, the CFI module creates a final output statement, which follows the defined command structure of the Grammar List.
- **Configuration File**
  The configuration file is required for configuring the behavior of the CFI module by adjusting the Priority List, Threshold List, feedback mode and other parameters.
- **Feedback**
  The feedback mode is enabled and adjusted by the configuration file. If activated, the validation algorithm conducts feedback from the user in order to ensure the correctness of the understood command. The command is illustrated at a device (e.g. screen) where the user is allowed to abort the analyzed command within a certain cancellation-timeout.
- **Heart Beat Filter**
  If a defined command structure includes a Heart Beat attribute (e.g. start or stop) the active state can be maintained by repeating the Heart Beat start attribute. During that repetition, the desired attribute is filtered to prevent an impact on the fusion algorithm.

After explaining the major functional blocks of the CFI module, the next section specifically focuses on an explanation of the fusion and interpretation workflow.

V. Algorithm

This section describes the relevant algorithm behind the CFI module. All incoming command attributes are validated, enriched with needed information such as timestamp and corresponding input channel and stored into the *Command Queue*. A prioritization algorithm sorts all commands in the queue and removes commands with lower priority. Lastly, the *Command Fusion Algorithm* is applied to form a full command statement.

A. Command Order Assumption

As defined in section III, the attributes can arrive in random order. This leads to complications in separating different commands. In the case of ambiguous command definitions, one attribute can belong to different commands. Therefore, the following assumptions were taken into account to provide a robust separation algorithm:

- Each command definition has to begin with a *Command Type* (e.g. `move, drill, sett`)
- Each received attribute has to arrive within a define time span to the last reception
- A command structure can be finished with a *Command End Key* (e.g. `go, ok, finish`)

B. Command Fusion Algorithm

The algorithm considered in this section combines all stored attributes to a final statement. The main challenge is a correct separation and validation of non-distinct commands. By applying the mentioned assumption, new commands will be accepted after exceeding the configured command times span or once a new Command Type is received.

Figure 4 shows the working principle of the command fusion. The upper part of the flowchart deals with the decision, whether the attribute to be analyzed belongs to the current command or to a new one. The decision is based on the command time span and a covered Command Type. If the attribute belongs to the same command, the Hypotheses List will be updated, otherwise, it will be renewed. With this new or updated Hypotheses List, the availability of the desired command can be analyzed. If no possibilities can be created from the collected attributes, we assume that the command is not defined and therefore not valid. But if possibilities are present in the Hypotheses List, the potential command is available. At this point, the specification of a fully received command has to be taken into account. Therefore, three conditions were established for defining a fully received command:

- Collected command attributes form a single hypotheses
- Configured command time span is exceeded
- Command End Key is received

If one of these conditions is satisfied, a fully received command is assumed. Otherwise, the algorithm remains in the waiting state until the command time span is exceeded.

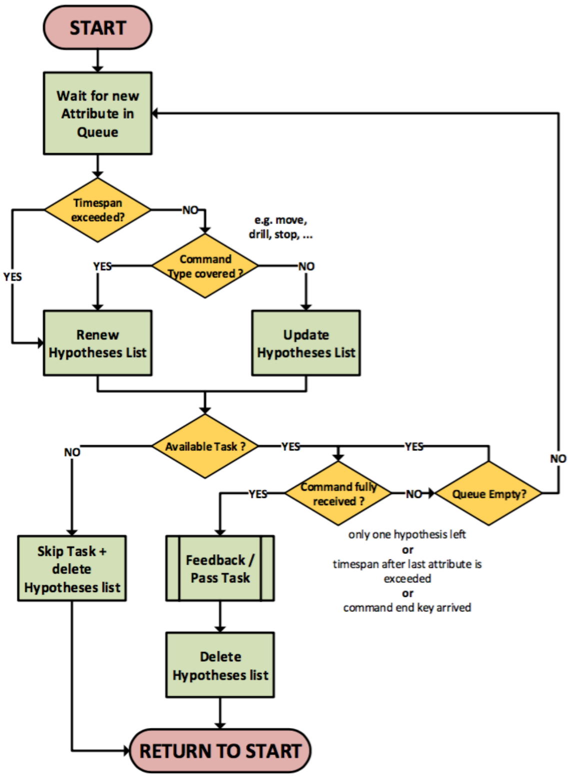

Fig. 4. Command fusion algorithm flowchart

VI. Testing and Environment

In order to test the CFI module, an environment is needed, including recognition systems as input channels such as speech, haptic, gesture and touch. Additionally, an execution module is needed for testing the impact and robustness of the output statement of the CFI module. Three different modalities over four recognition systems were used which includes the *CMUSphinx*[1] as an offline speech recognition system for low-resource platforms, the *Microsoft Kinect* and *Leap Motion*[2] as commercial gesture capturing devices specialized for hand or body motions and the *FBRT*[3] as a java based runtime environment of IEC 61499 function block for graphical user interfaces. The output of the CFI module is received by either an Universal Robot simulation (UR)[4] software URSim or the robot control unit of the real UR. All developed modules are executed using *FORTE*[5], which is an IEC 61499 compliant runtime environment. By combining these recognition and control systems, a testing framework can be built.

For testing the CFI module a representative use case was established for programming a robot. Therefore, two modes were established:

- **Teach Mode:**
  Robot can be moved and digital IOs can be set by using defined commands (e.g. `move up`, `gripper close`). All commands are stored with the corresponding timestamp and form a kind of a recipe. In this mode, all defined commands are available.
- **Replay Mode:**
  A built recipe will be executed to replay the robot movement. In this mode, only management and security commands are available.

A *Mode Manager* was developed for switching between these two modes and configuring the CFI module accordingly using management command (e.g. `record`, `replay`, `default`). All other commands are used for controlling a robot.

Test Case

Using the established test environment, a simple pick and place application was conducted in order to evaluate the usability and robustness of the system. A Universal Robot UR10 CB3 was equipped with a vacuum gripper and integrated in a worktable. The user had the task to use the available modalities (speech, hand gesture, body gesture and touch) in order to form statements to navigate the robot to a workpiece, grip it with the vacuum gripper and place it somewhere else on the table. Afterwards, the mode manager was used to record this procedure for repetition.

Based on this use case, a video was recorded, which explains the multimodal command fusion and interpretation functionalities. The clip can be retrieved from the following link[6].

VII. Discussion

The developed CFI module represents a technology to fuse and interpret predefined command structures from several input modalities. By configuring the module, the different input channels can be weighted via a priority key in order adapt the fusion process to the individual needs. Unreliable sources such as gesture and speech recognition can produce contradicting and invalid statements which were taken into account. Additionally, these sources often provide a confidence level related to the understood command, which are included into the fusion process as well. In order to provide a highly flexible module, all defined command structures can be adjusted during the execution time.

Systems which integrate a CFI module allow the non-expert users to use several modalities for programming, controlling or adjusting a machine. Due to the definable Grammar, which can be adjusted during runtime, the valid commands can be adapted according to the operators needs or experience level and create an intuitive interface. The command order assumption (discussed in section V-A), i.e. a

[1] `https://cmusphinx.github.io`
[2] `https://www.leapmotion.com`
[3] `https://www.holobloc.com`
[4] `https://www.universal-robots.com`
[5] `https://www.eclipse.org/4diac/en_rte.php`

[6] `https://youtu.be/AbJ8VaxxwzI`

key command (e.g. `move, drill, set`) has to be the first attribute, did not prove to be a limitation of the usability of the system. As known from different commercial recognition system, key words are used to indicate a command. Therefore, ordinary users natively initiates a operation with this key command and thus increases the usability of the system.

Using individual tailored Grammar for every user, no expert is needed for programming a machine. This increases the flexibility of a system, because highly qualified technicians are often temporarily unavailable.

The test use case indicated, that the used recognition systems were not accurate enough to gain a high usability of the system. The speech recognition was very sensitive to noise and therefore not usable during the active vacuum gripper. The generic command structures enabled programming up to a certain granularity. Thus, high accuracy programming turned out to be challenging. Parameterized commands (e.g. move to coordinates, move distance) are part of future work to increase the usability of the system. After a procedure was recorded properly, the replay functionality of the Mode Manager imitated the movements accurately.

VIII. Future Work

In order to gain a higher stability of generating complex command structures, investigations into new recognition systems are needed. Furthermore, the modalities should be evaluated regarding their usability of generating robot relevant commands. Additionally, the modalities have to be analyzed regarding the possibility of generating parameters for commands, e.g. coordinates, velocities or identifiers.
Since the CFI module was only tested by the developer team, new testing scenarios with ordinary users have to be established to test the stability and user-friendliness of the CFI system. Based on a survey of these users, the used robot commands can be evaluated in terms of intuitiveness.

IX. Acknowledgment

This work was funded by the project European research project SYMBIO-TIC[7] (European Commission; Horizon 2020 program, under grant agreement 637107).

References

[1] W. H. Adams, G. Iyengar, C.-Y. Lin, M. R. Naphade, C. Neti, H. J. Nock, and J. R. Smith, "Semantic indexing of multimedia content using visual, audio, and text cues," vol. 2003, no. 2, pp. 1–16.

[2] B. Akan, A. Ameri, B. Çürüklü, and L. Asplund, "Intuitive industrial robot programming through incremental multimodal language and augmented reality," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3934–3939.

[3] E. A. Ameri, B. Akan, and B. Çürüklü, "Incremental multimodal interface for human robot interaction," in *2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–4.

[4] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia systems*, vol. 16, no. 6, pp. 345–379, 2010.

[5] B. Burger, I. Ferrané, F. Lerasle, and G. Infantes, "Two-handed gesture recognition and fusion with speech to command a robot," vol. 32, no. 2, pp. 129–147.

[6] J. Cacace, A. Finzi, and V. Lippiello, "Robust multimodal command interpretation for human-multirobot interaction." in *AIRO@ AI* IA*, 2017, pp. 27–33.

[7] A. D'Ulizia, F. Ferri, and P. Grifoni, "Generating multimodal grammars for multimodal dialogue processing," vol. 40, no. 6, pp. 1130–1145.

[8] G. A. Farulla, L. O. Russo, V. Gallifuoco, and M. Indaco, "A novel architectural pattern to support the development of human-robot interaction (HRI) systems integrating haptic interfaces and gesture recognition algorithms," in *2015 IEEE Computer Society Annual Symposium on VLSI*, pp. 386–391.

[9] M. Gandetto, L. Marchesooti, S. Sciutto, D. Negroni, and C. S. Regazzoni, "From multi-sensor surveillance towards smart interactive spaces," in *2003 International Conference on Multimedia and Expo, 2003. ICME '03. Proceedings*, vol. 1, pp. I–641–4 vol.1.

[10] R. Gomez, K. Nakamura, T. Kawahara, and K. Nakadai, "Multi-party human-robot interaction with distant-talking speech recognition," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 439–446.

[11] H. Holzapfel, K. Nickel, and R. Stiefelhagen, "Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3d pointing gestures," in *Proceedings of the 6th International Conference on Multimodal Interfaces*, ser. ICMI '04. ACM, pp. 175–182.

[12] L. Kessous, G. Castellano, and G. Caridakis, "Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis," vol. 3, no. 1, pp. 33–48.

[13] E. Lakshantha and S. Egerton, "A diagrammatic framework for intuitive human robot interaction," *Journal of Ambient Intelligence and Smart Environments*, vol. 8, pp. 21–33, 01 2016.

[14] H. Liu, T. Fang, T. Zhou, Y. Wang, and L. Wang, "Deep learning-based multimodal control interface for human-robot collaboration," *Procedia CIRP*, vol. 72, no. 1, pp. 3–8, 2018.

[15] A. P. Loh, F. Guan, and S. S. Ge, "Motion estimation using audio and video fusion," in *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, vol. 3, pp. 1569–1574 Vol. 3.

[16] S. Oviatt, A. DeAngeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human-computer interaction," in *Referring Phenomena in a Multimedia Context and Their Computational Treatment*, ser. ReferringPhenomena '97. Association for Computational Linguistics, pp. 1–13.

[17] F. C. Pereira and D. H. Warren, "Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks," *Artificial intelligence*, vol. 13, no. 3, pp. 231–278, 1980.

[18] C. P. Quintero, R. Tatsambon, M. Gridseth, and M. Jägersand, "Visual pointing gestures for bi-directional human robot interaction in a pick-and-place task," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 349–354.

[19] S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno, "An extensible architecture for robust multimodal human-robot communication," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2208–2213.

[20] J. Scholtz, "Theory and evaluation of human robot interactions," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*. IEEE, 2003, pp. 10–pp.

[21] H. Shimazu, S. Arita, and Y. Takashima, "Multi-modal definite clause grammar," in *Proceedings of the 15th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1994, pp. 832–836.

[22] O. M. I. E. Sucar, S. H. Aviles, and C. Miranda-Palma, "From HCI to HRI - usability inspection in multimodal human - robot interactions," in *The 12th IEEE International Workshop on Robot and Human Interactive Communication, 2003. Proceedings. ROMAN 2003*, pp. 37–41.

[23] M.-T. Yang, S.-C. Wang, and Y.-Y. Lin, "A multimodal fusion system for people detection and tracking," vol. 15, no. 2, pp. 131–142.

[7] `http://www.symbio-tic.eu`