CLARC: a Robotic Architecture for Comprehensive Geriatric Assessment

Antonio Bandera, Juan Pedro Bandera, Pablo Bustos, Luis V. Calderita, Álvaro Dueñas, Fernando Fernández, Raquel Fuentetaja, Ángel García-Olaya, Francisco Javier García-Polo, José Carlos González, Ana Iglesias, Luis J. Manso, Rebeca Marfil, José Carlos Pulido, Christian Reuther, Adrián Romero-Garcés, Cristina Suárez

***Abstract*—Comprehensive Geriatric Assessment (CGA) is an integrated clinical procedure to evaluate frail old people status and create therapy plans to improve their quality and quantity of life. In this paper we present CLARC, a mobile robot able to receive the patient and his family, accompany them to the medical consulting room and, once there, help the physician to capture and manage their data during CGA procedures. The hardware structure of CLARC is based on a robotic platform from MetraLabs. The software architecture of the system incorporates a deeply tested framework for interactive robots. This framework, by encoding the whole CGA session using Automated Planning, is able to autonomously plan, drive, monitor and evaluate the session, while also managing robot navigation and data acquisition. CLARC incorporates a series of sensors allowing to collect data automatically, using non-invasive procedures. The healthcare professional can use the platform to automatically collect data while addressing other tasks such as personal interviewing, data evaluation or care planning. First trials will be carried out in hospitals in Seville and Barcelona in June and July 2016, respectively.**

***Index Terms*—Gerontechnology, Automated Planning, Intelligent Robots, Medical Robotics, Human-Robot-Interaction**

I. Introduction

COMPREHENSIVE Geriatric Assessment (CGA) is a powerful procedure for the evaluation and treatment prescription of frail older people. CGA first evaluates the patient's clinical, functional, environmental and psychosocial status, and compares its temporal evolution. Then, an overall treatment and follow-up plan is prescribed. CGA is an interdisciplinary effort involving the coordination of different medical staff, which is being carried out all over the world, with the aim of increasing both the quality and quantity of life of frail adults. Some of the benefits of CGA are improving the diagnostic, creating right, customized and proportional therapeutic plans, increasing functional autonomy, and also reducing complications during hospitalizations and mortality. Giving the aging of the world population, with about 810 million people over 60 in 2012, which is expected to grow to more than 2 billions in 2050, CGA importance, and costs related to it, are by no doubt going to be increased.

CGA procedures vary from hospital to hospital but in general they are carried out every 6 months, involve both patients and relatives, and are made of 3 different types of activities: clinical interview, multidimensional assessment and customized care plan. During the clinical interview patient and relatives comment with the physicians about the elder health problems. Next, multidimensional tests are performed to evaluate the overall patient status. In questionnaire-based tests, the patient or relatives answer some questions about patient daily life and his/her ability to perform some activities without help. Depending on the answers a score is given to the patient. The *Barthel* Index test [9] is an example of such tests. Another type of tests involve the observation of the patient performing some activities, like in the *Get Up and Go* test [11], where the patient is asked to get up from the chair, walk for a few meters and come back to the original place. Finally, based on the evidences gathered during the two previous phases and the patient's evolution from the last CGA session, physicians create a personalized care plan to be followed until the next review. A typical CGA session lasts about 3 hours, and there are many parts that could be parallelized or automatized, especially during the multidimensional assessment. For example, some activities must be performed individually by both patient and relatives, so they can be run simultaneously at different rooms, and some tests do not need the presence of a physician to be performed.

In this paper we present the architecture and preliminary implementation of CLARC[1], an autonomous robotic solution to support CGA. It provides a web graphical interface allowing the physicians to specify the tests to be answered by a patient during a CGA session. Once the multidimensional assessment is designed, the robot is able to perform and mark the tests by interacting or observing the patient, store the results, and maintain a record the physician can use to design the treatment plan. CLARC, using both speech recognition and touch-screen interaction, is able not only to automatically collect data from patients and relatives by conducting questionnaires and interview-based tests, but it is also able to perform direct observation (face expressions, body pose and motion, and speech parameters), needed in observation-based tests. By encoding

Antonio Bandera, Juan Pedro Bandera, Rebeca Marfil and Adrián Romero-Garcés are with University of Málaga. E-mail: {ajbandera, jpbandera, rebeca, argarces}@uma.es

Fernando Fernández, Raquel Fuentetaja, Angel García-Olaya, Francisco Javier García-Polo, Ana Iglesias, José Carlos González and José Carlos Pulido are with University Carlos III de Madrid. E-mail: {ffernand, rfuentet, agolaya, fjgpolo, aiglesia, josgonza, jcpulido}@inf.uc3m.es

Pablo Bustos, Luis V. Calderita and Luis J. Manso are with University of Extremadura. E-mail: {pbustos, lvcalderita, lmanso}@unex.es

Christian Reuther is with MetraLabs. E-mail: christian.reuther@metralabs.com

Álvaro Dueñas and Cristina Suarez are with Hospital Universitario Virgen del Rocío. E-mail: alvaro.duenas.ruiz@hotmail.com and cristina.suarez.exts@juntadeandalucia.es

[1]http://echord.eu/essential_grid/clark/

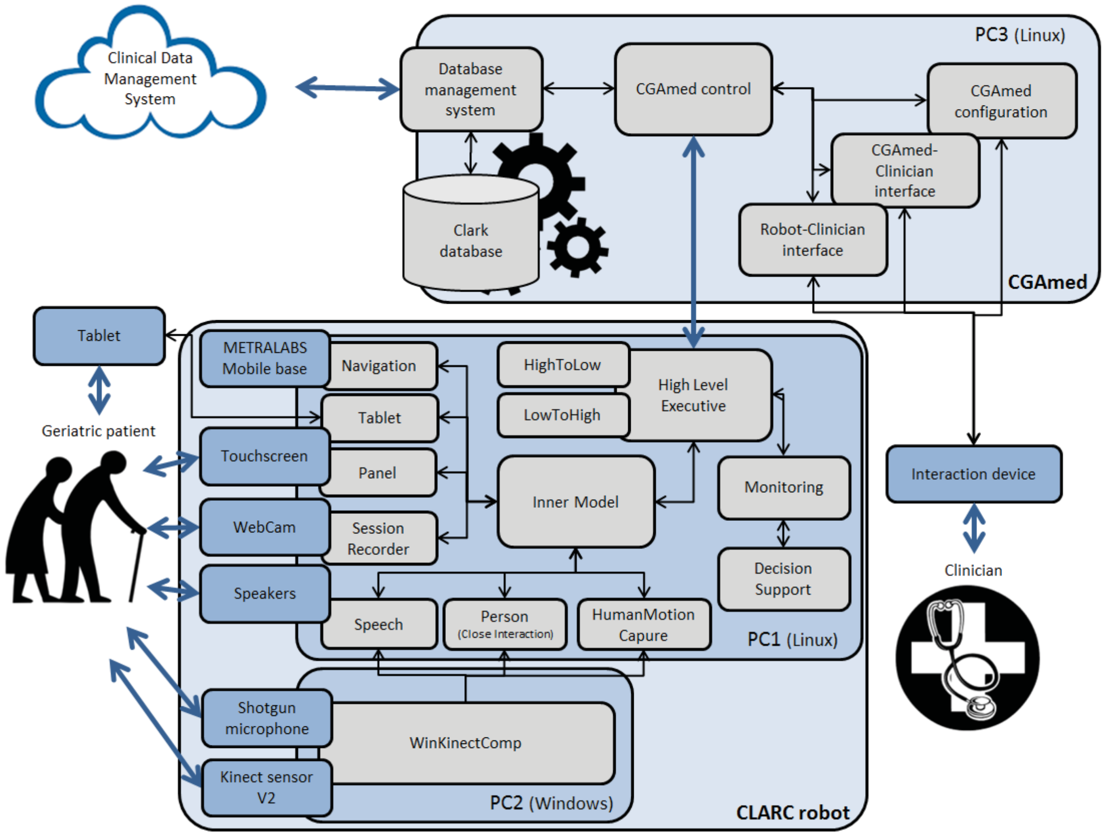

Fig. 1. CLARC conceptual architecture

the whole assessment cycle using Automated Planning it is able to autonomously work, without any help, adapting the test to patient behavior and managing the unexpected events that can appear during a session.

The remaining of the paper is structured as follows: Section II presents the overall system architecture, describing the robot, the human-robot interaction modules, the deliberative module, the interface and the integration with the Clinical Data Management System. An example of a typical CGA session using CLARC is shown in Section III. The current status of the system is described in Section IV, while Sections V and VI describe the related work, and the future developments and the conclusions, respectively.

II. System Architecture

The architecture of the CLARC system is shown in Figure 1. CLARC is composed of several modules, running either on the robot or on external PCs, including the clinician's PC. A total of 3 computers support the architecture, two of them are part of the robot and are used to control it and to interact with patients. The third computer is placed outside the robot and supports the database system and the system-clinician interface, what we called CGAmed.

From a conceptual point of view the system can be divided into three main components; the Robot, the Cognitive Architecture and the CGAmed software. The robot is a mobile platform, based on the MetraLabs SCITOS G3, and equipped with extra sensors to be able to seamlessly perform and record tests and interact with patients and relatives. The cognitive architecture, running on-board the robot, provides it with the needed intelligence to perform its tasks. CGAmed supports the interface of the clinician both with the robot (to configure for example the tests to be performed) and with the generated data (patient profile, recorded sessions, tests marks, etc.). Connection of the robot and the CGAmed is done in two ways. The main link connects the *High Level Executive* of the robot's cognitive architecture to the *CGAmed Control* module. The later commands the former to switch on the remaining robot modules and transfers the information about the tests to be performed. All the configuration information and the results of the session travel through this connection. More details are provided in Section III. Although it is not shown in the figure, there is also a second direct link between the *Session Recorder*

Fig. 2. CLARC robot prototype design

module of the robot and the database.

A. The Robot

A MetraLabs SCITOS G3 platform is being adapted to meet the requirements of the defined use case. The outer shell is currently being redesigned to accommodate the new sensors and to customize it for the specific CGA needs. The robot's locomotion is based on a differential drive system consisting of two powered wheels and a caster wheel for stability. This enables the robot to rotate on the spot and drive at speeds of up to 1 m/s, if necessary. The platform contains a 40Ah lithium battery which allows for up to 18 hours of autonomous operation, and can be recharged fully within 4 hours. A safety bumper socket sensor around the outer perimeter of the robot's shell is used to prevent the robot from exerting force against animate or inanimate objects. The platform is fitted with a LIDAR sensor for localization, navigation and obstacle avoidance.

The SCITOS G3 platform is extended with an extensive human-machine-interface, consisting of a Microsoft Kinect V2 sensor, a shotgun microphone, a touch screen and speakers for multi-modal human-robot interaction, as well as a web cam for recording the sessions. The system is also provided with a external tablet mirroring the touch screen, that the patient can use to interact with the robot if desired. Figure 2 shows a prototypical adaption of the SCITOS G3 platform for the CLARC use case.

B. The Cognitive Architecture

CLARC robot benefits from using the RoboCog [1] cognitive software architecture to control its behaviour. RoboCog proposes a distributed architecture, where action execution, simulation, and perception are intimately tied together, sharing a common representation, the Inner Model In the CGA scenario, this internal representation of the robot, the patient and any other significant event captured from the outer world, is the central part of the architecture for action control. The rest of task-solving elements of RoboCog (the Panel, Tablet, Speech, etc. see Figure 1) use this central representation to share data at different abstraction levels, to get information about the user's state and to plan next actions.

The robot's course of action emerges from the activity of several networks of software components (called *compoNets*), which are connected to the Inner Model through a specific component (called the *agent*). Each compoNet is currently able to endow the robot with a specific ability. Some compoNets are connected to sensors, and they process their raw data to enrich the inner representation with fast perceptions. Some other ones are connected to actuators, which allow the robot to interact with its environment. It is usual that a compoNet manages either sensors or actuators, but this is not a requisite. For instance, the PELEA Deliberative compoNet, in charge of providing the planning, monitoring and high-level learning abilities, works over the data stored in the Inner Model or the CGAmed central server.

All the architecture runs in a Linux computer, interacting via the shared inner representation. That means that there is no direct connection between compoNets, which continuously check the data contained in the Inner Model, update it and act in consequence. Figure 1 also shows the existence of a second PC within the robot. It runs the *WinKinectComp* component, which is in charge of handling the data coming from the Kinect sensor and the microphone. It processes and provides a continuous stream of information to those compoNets that need the data related to the person in front of the robot, namely *Speech* recognition, *Person* (Close Interaction) and *HumanMotionCapture*.

1) Patient-Robot Interaction: The so called low-level components of the robot provide the necessary functionality to perform the Patient-Robot interaction. They are driven by the changes on the inner representation, which could be provoked by the Deliberative compoNet (see Section III) or by an incoming perception. Furthermore, the results of the actions are also added to the Inner Model, allowing the Deliberative module to reason about them. The collection of compoNets initially planned to be included within the software architecture includes:

- The *Panel* and *Tablet* compoNets, which manage the interaction with the patient via the touchscreen and the tablet, respectively. The tablet is specially useful in tests, as the *Mini-Mental* one, where there are questions where the patient is asked to hand-write.
- The *Speech* compoNet manages the verbal communication with the patient, being able to both speak and hear to the patient. Patient-robot interaction is redundant in the sense that information is usually shown to the patient using text and voice simultaneously, and the answer can be received also by voice or by selecting on the touchscreen. Accessibility issues have been taken into account to customize both the information provided and the feedback modes to the particular needs of frail older people. Both verbal and graphical interfaces are multi-language.
- The *Person* compoNet is in charge of detecting and

tracking the person sitting in front of the robot (upper-body motion capture, including hands and fingers).

- The *HumanMotionCapture* compoNet is the responsible of capturing the whole motion of the person, providing information about all joints. It is necessary for addressing tests such as the *Get Up & Go*.
- The *Session Recorder* component, as said, manages the data of the on-board webcam to record both the audio and video of the session. The video is temporarily annotated by the deliberative module and stored into the database. That way the clinician can review the video of any session and analyze the patient behavior.

2) Autonomy: The Deliberative module is based on the PELEA [8] Automated Planning and Learning architecture. Complementing the low-level detection of exogenous events, the use of Automated Planning allows to control the robot, providing it with full autonomy. Automated Planning (AP) aims to find an ordered sequence of actions that allows the transition from a given initial state to a state where a series of goals are achieved. The use of AP for robotic control in clinical applications has been tested in previous works [5], where it demonstrated its ability to conduct rehabilitation sessions with children suffering from cerebral palsy in a fully autonomous way. The Planning Domain Definition Language (PDDL) [12] is used to describe the environment and the actions the robot can perform, both in terms of predicate logic. Describing an action in PDDL is as simple as enumerating its preconditions and effects. Preconditions are the facts that must be true in a certain state for the action to be applicable, while effects are the new facts appearing and the facts that are no longer true after the action is applied. The environment is also modeled using predicate logic to describe the objects, their properties and relationships. Adding new actions or facts or changing the existing ones is done easily by just editing a text file. Figure 3 shows an example of an action for the *Barthel* test. Several instances of this action are executed at the introductory part of the test, since introducing the test implies to execute several introductory acts. This is the general action for all of them. Specifically, for the introduction labeled as `?i`, this action can be applied only if there is no external cause that prevents continuing the test (predicate `can_continue`), the robot has been introduced (predicate `finished_introduce_robot`), and the introduction `?i` is the next one, following the test order (next two preconditions). The parameter `?pause` represents the pause in seconds that the robot should perform after executing the action. The effects of the action are that this part of the introduction has finished `(introduction_finished ?i)` and the system is ready for the next part of the introduction, if any.

At the beginning of the execution the Deliberative module receives the goals to pursue from the CGAmed module (for example: perform to patient Paula Smith a *Barthel* test in room A and a *Mini-Mental* [4] test in room B starting at 9:30 am). Taking into account these goals and the description of the environment contained in the Inner Model, a new planning problem is created and a plan is returned. The plan includes the high-level actions the robot has to perform to achieve the goals, for example greet the patient, introduce the test to be performed, say the first question, wait for the answer, etc. Non-expected states and exogenous events are contemplated, so the robot is able to, for example, repeat a question if the patient does not answer, ask him/her to seat if he/she stands up or call the physician if something wrong is detected. Figure 4 shows an example of the first part of a plan generated for the *Barthel* test. In this plan, the actions contain labels, as `INTRO1` or `Q1_S1`, that represent specific acts for the robot. The action 1 refers to an interruption of the test since the patient has not been detected, so the robot should call him/her. The next action is executed when the cause of the interruption has been solved, which allows to continue with the test.

```
(:action introduce-test
 :parameters (?i - introduction ?p - pause)
 :precondition (and
        (can_continue)
        (finished_introduce_robot)
        (is_test_introduction ?i)
        (= (current_introduction)
           (test_introduction_position ?i))
        (after_pause ?i ?p) )
 :effect (and
         (introduction_finished ?i)
         (increase (current_introduction) 1)))
```

Fig. 3. Example of a PDDL action for the *Barthel* test

```
0: (CONFIGURE-TEST BARTHEL SPANISH PATIENT PRESENT)
1: (INTERRUPT BARTHEL PATIENT_ABSENT ROBOT_CALL_PATIENT)
2: (RESTORE-FROM ROBOT_CALL_PATIENT PATIENT_ABSENT)
3: (INTRODUCE-ROBOT ROBOT_PRES1 THE_ROBOT PAUSE_0SG)
4: (INTRODUCE-TEST INTRO1 PAUSE_1SG)
5: (INTRODUCE-TEST INTRO2 PAUSE_1SG)
6: (INTRODUCE-TEST INTRO3 PAUSE_1SG)
7: (INTRODUCE-TEST INTRO4 PAUSE_1SG)
8: (START-QUESTION Q1_S1 Q1 PAUSE_1SG)
9: (SHOW-QUESTION-OPTION Q1_O1 Q1_O1 Q1 FIRST PAUSE_1SG)
10: (SHOW-QUESTION-OPTION Q1_O2 Q1_O2 Q1 FIRST PAUSE_1SG)
11: (FINISH-QUESTION Q1_E1 Q1 PAUSE_10SG)
12: (ASK-FOR-ANSWER Q1_A1 Q1)
13: (RECEIVE-ANSWER Q1_A1 Q1 DUR_6SG)
14: (FINISH-ASK-ANSWER-SUCCESS Q1)
15: (MAKE-QUESTION-TRANSITION Q1_T Q1 PAUSE_4SG)
...
```

Fig. 4. Example of the first part of a possible plan for the *Barthel* test

The plan involves changing the Inner Model for provoking the required response from the low-level components or to read this representation for determining the current state of the world. Thus, actions like switch the camera on, say a given phrase, show a given text at the touchscreen, receive a verbal answer, etc. are translated to inner events on the model that the rest of compoNets must solve. These events are complementary to other external ones such as a motion of the person's face, which is reactively solved by the Person compoNet. Other actions such as determining the position of the patient's arms are solved by examining the Inner Model. For example if the patient is absent as in the plan above, the speech component will receive the "call the patient" action. The Person compoNet, which was the responsible of informing of the absence, will detect whether the patient comes back. Once the patient is again seated, Person compoNet will add the information to the Inner Model, so the Deliberative module can continue with the next action of the plan.

Following the PELEA structure, the components of the Deliberative module are the following:

- The *High Level Executive* (HLE) module receives the goals from the CGAmed system and invokes the Monitoring module to get a plan achieving them. Then it takes the first action of the plan and invokes the HighToLow module to decompose it into low-level actions understandable by the low-level components of the robot. These actions are then inserted into the Inner Model and executed by the low-level components. Those components update the Inner Model with the results of the actions. HLE looks at the changes in the Inner Model and, after a conversion to high level knowledge performed by LowToHigh, sends them to Monitoring that checks whether the plan is executing conveniently.
- The *Monitoring* module, maintains a high level model of the environment and is in charge of invoking the Decision Support module if any deviation in the execution of the plan arises. It detects for example that the user has not answered a question or is not facing the robot and tries to find alternate plans to solve the problem found.
- The *Decision Support* module creates a plan starting from the current state, the goals to be achieved, the possible states of the world and the description of the changes the actions produce in the world state, all of them expressed in PDDL. To create the plan it invokes an automated planner that returns the sequence of actions achieving the goals. Using PDDL allows the planner to be changed seamlessly, thus benefiting from any improvement in the planning community.
- The *HighToLow* module converts the high level actions of the plan created by the Decision support module into low level actions that can be included into the Inner Model.
- The *LowToHigh* module converts the information contained in the Inner Model, which represents knowledge in the form of binary predicates (see Section II-B4) into n-ary predicates that the Monitoring module uses to reason about the correctness of the execution of the plan.

3) Navigation: Autonomous navigation is realized using MetraLabs' proprietary navigation software CogniDrive. CogniDrive consists of modules for localization, navigation and obstacle avoidance. The localization module uses an adaptive particle filter to track multiple position hypotheses at the same time, and therefore allows for accurate localization even when faced with ambiguity in the sensor data. The navigation module uses an adaptive derivative of the A* planning algorithm to generate global paths, which are adapted on a local scale by the immediate temporal history of local obstacle measurements. Local scale path planning and obstacle avoidance is addressed using the established Dynamic Window Approach (DWA). CogniDrive allows the definition of no-go areas that are to be avoided in local and global path planning, as well as the definition of speed areas which can limit the robot's movement speed in critical environments. MetraLabs has deployed over 200 autonomous mobile robots using CogniDrive, with over 60,000km of autonomous driving experience in complex and crowded environments.

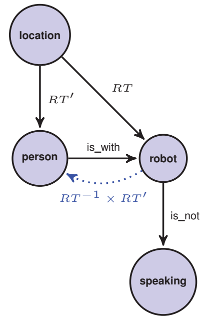

Fig. 5. Unified representation as a multi-labeled directed graph. Edges labeled as `is_with` and `is_not` denote logic predicates between nodes. Edges starting at `location` and ending at `person` and `robot` are geometric and encode a rigid transformation (RT' and RT respectively) between them. Geometric transformations can be chained or inverted to compute changes in coordinate systems.

4) The Inner Model: The Inner Model is a multi-labeled directed graph which holds symbolic and geometric information within the same structure. Symbolic tokens are stated as logic attributes related by predicates that, within the graph, are stored in nodes and edges respectively. Geometric information is stored as predefined object types linked by 4×4 homogeneous matrices. Again, they are respectively stored as nodes and edges of the graph. Figure 5 shows one simple example. The `person` and `robot` nodes are geometrical entities, both linked to the `location` (a specific anchor providing the origin of coordinates) by a rigid transformation. But, at the same time that we can compute the geometrical relationship between both nodes ($RT^{-1} \times RT'$), the `person` can be located (`is_with`) close to the `robot`. Furthermore, an agent can annotate that currently the `robot is_not speaking`.

C. The CGAmed module

The CGAmed module manages the communication of the clinician and the robot and provides access to the data stored in the platform. Its components are:

- The *Robot Clinician Interface* provides the clinician with the tools needed to configure a CGA session and to monitor it in real time. It is developed as a web interface that can be accessed from the clinician's computer or from a tablet. Figure 6 shows the interface to schedule a CGA test for a patient. The clinician can select a patient from the list of registered ones and schedule a test for him/her, specifying the time and location where it will take place. Additional parameters, as for example asking the patient about his/her state 6 months ago instead of today, can be also configured. Figure 7 shows the monitoring screen. This interface allows the clinician to start, pause and remotely monitor a CGA session performed by the robot. A live video of the session is shown, as it is recorded by the *Session Recorder* module. Also the log of the session is shown on the right upper

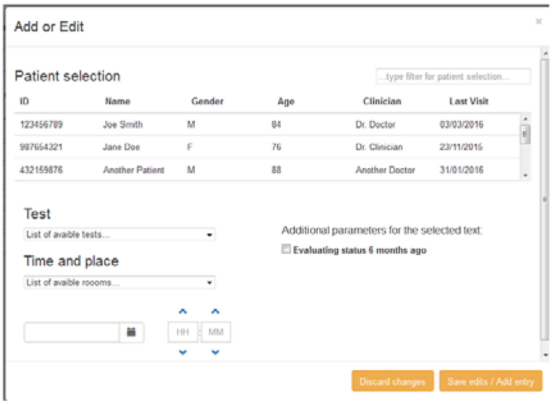

Fig. 6. Robot Clinician Interface: creating a new test

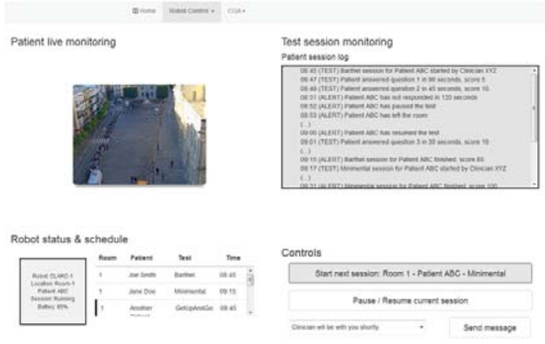

Fig. 7. Robot Clinician Interface: monitoring a live CGA session

part of the screen. The clinician can see a summary of the robot status and its schedule, including next tests to be performed by the robot. Finally, the interface provides the doctor with the ability to use the robot to interact in a limited way with the patient, by sending a series of predefined messages that will be reproduced by the robot speakers and touchscreen.

- The *CGAmed Clinician Interface* allows the clinician to access the clinical data stored into the system. Once a patient is selected, demographic data are shown, along with a list of past tests, including their scores and any additional information deemed important. The clinician can edit the tests to modify the automatic score set by the robot, viewing the video recorded for each part of the test, or comparing the videos for the same parts of tests performed in different dates to assess the patient evolution with time.
- The *CGAmed Configuration* allows the clinician to configure the system. Parameters as the system language can be set.
- The logic under the three former modules is provided by the *CGAmed control*, which is also the gateway to the rest of the system and to the database. This module communicates with the *High Level Executive* sending the information about the tests to be performed and receiving the monitoring data. It also controls the *Data Base Management System* and its integration with the Clinical Data Management System of the Hospital, via the Clinical Document Architecture (CDA) of HL7.

III. A CGA SESSION USING CLARC

From a clinician point of view, a CGA session using CLARC begins by login into the CGAmed web interface and creating the list of tests to be performed next (see Figure 6). When the patient or relative is ready to answer the test, the clinician press the start test button on the computer and accompanies him/her to the room where the robot is (in a near future we plan the robot to autonomously accompany the patient to the room). Once the robot detects the patient at the room, the test begins. The robot starts greeting the patient and explaining the purpose and structure of the test. If it is a question-based test, questions are presented both orally and on the touchscreen and patients can answer by voice or by selecting the right answer on the screen. In the case of an observation test, the robot asks the patient to perform the required activities and monitors its performance using the Kinect sensor. In both cases, the system automatically marks the patient performance and stores the scores into the database. The monitoring abilities of the software architecture allow CLARC to ask for help to the medical expert if needed and to recover from unexpected situations as the patient leaving the room, asking for help or not being able to give an appropriate answer for a question.

Meanwhile the clinician can monitor the session from his/her office (see Figure 7) and change the automatically set scores once the test is finished. Both scores are kept for tracking purposes (see Figure 8). Whichever the type of the test, the whole patient-robot interaction is video and audio recorded by the web cam and temporarily annotated by the Deliberative module. This allows the clinician to offline review the tests and to go directly to the video recording of any specific part of them, even doing side-by-side video comparison of the performance of the patient with that of previous CGA sessions.

From the system point of view, once the physician presses the start button, the tests to be performed and their configuration parameters (patient, room, etc.) are sent to the Deliberative component that creates a plan to fulfill them. It then commands the low-level components to perform the desired activities (introduce the robot, introduce the test, ask for a question, wait for an answer, monitor the patient movements...) by doing appropriate changes in the Inner Model. The low level components perform their tasks and update the Inner Model with the results. In turn, the Deliberative component sends updates about the current state to the CGAmed control module. Figure 9 shows a simplified sequence diagram of a use case where a clinician uses CLARC to perform a patient evaluation based on *Barthel* and *Mini-Mental* tests. It is a simplification since the low-level components of the architecture are not included, so many steps are skipped.

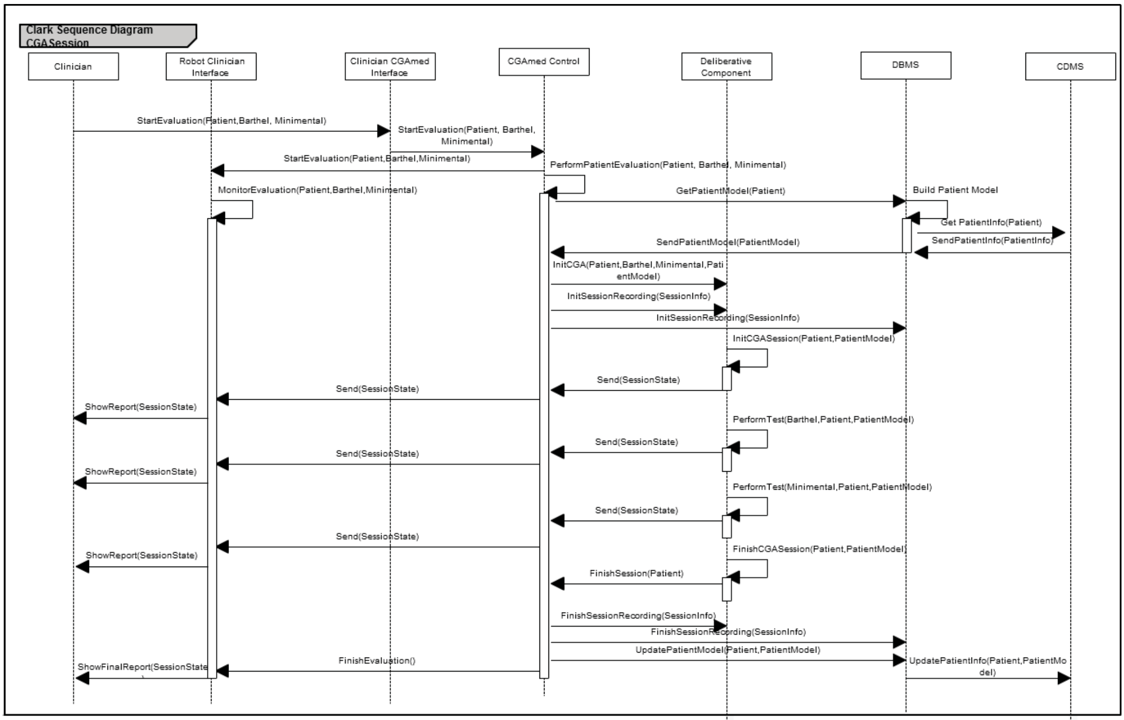

Fig. 9. A sequence diagram showing the interactions between components for a simplified CGA based on a *Barthel* and a *Mini-Mental* test

Fig. 8. Robot Clinician Interface: reviewing and editing a CGA session

IV. Current status

CLARC is still under development and the goal is to have a commercial system by 2018, supporting several CGA tests with total autonomy. Currently a first fully functional version of most of the components has been developed and the PDDL descriptions for the *Barthel*, *Mini-Mental* and *Get Up and Go* tests have been created. *Barthel* and *Mini-Mental* are mainly questionnaire-based tests and their scoring is done automatically by the system, although the physician can change the scores anytime. The *Get Up and Go* test needs the evaluation of the patient's body motion, so its scoring is done manually by the clinicians. A first prototype of the CGAmed interface is also ready. The robot is able to perform a complete *Barthel* test, considering the most frequent errors that could appear, as the patient not answering a question or leaving the room.

Trials with volunteer patients for the three previous tests will be conducted at Hospital Universitario Virgen del Rocío in Seville, Spain, in June 2016. In July, the system will be presented to clinicians of Hospital Sant Antoni Abat (Vilanova, Spain), where further tests will be performed to improve the system functionality and to adapt it to user preferences.

V. Related work

To our knowledge there is no currently any robotic system aimed to assist clinicians in performing CGA. CLARC was born in response to a competitive call launched by the The European Coordination Hub for Open Robotics Development (ECHORD++) project[2], where it competes against two other approaches ARNICA and ASSESSTRONIC. Two of these three approaches will continue to be funded after the first trial in Vilanova, July 2016. ARNICA[3] uses a Kompai robot to perform CGA, but no Artificial Intelligence capabilities seems

[2]More info about the project can be found at http://echord.eu
[3]http://echord.eu/essential_grid/arnica/

to be provided. ASSESSTRONIC[4] also focuses more on the Human-Robot Interaction, including non-verbal interaction, than in the system intelligence.

Most systems designed to direct questionnaire filling tasks do not rely on the exclusive use of natural interaction channels, and force the user to employ a keyboard or a mouse device [6]. However, recent proposals in assistive robotics deny the use of these interfaces and focus on the use of artificial conversational systems, touch screens or a combination of both. One interesting example is proposed in the ALIAS Project[5]. Our proposal follows the same approach and uses only natural interaction channels (i.e. voice and touch). To our knowledge, these multimodal interfaces have not yet been applied for automated CGA processes.

Gait analysis, on the other hand, has been traditionally achieved using invasive approaches, such as marker-based motion capture systems. These systems are still the most popular option for medical or rehabilitation scenarios, but require a controlled environment and the user to wear specific markers [13]. One of the challenges for the current proposal is to effectively capture human motion using only the sensors mounted in the robot. Such a system will reduce setting up times and will be more comfortable for the user.

CLARC deliberative system can be considered a successor of NaoTherapist [5][10], a robotic platform for rehabilitation of children with cerebral palsy. NaoTherapist is able to autonomously plan, execute and monitor a rehabilitation session, made of different exercises the robot shows and the child imitates. While monitoring the exercises the system is able to fill some of the items of the QUEST [3] test. The gesture monitoring capabilities of NaoTherapist are somehow limited and there is no real verbal robot-child interaction, despite the robot is able to speak to encourage the kid. Robots and sensors, like Kinect, have been also used for rehabilitation sessions including patient monitoring and evaluation [7]. But the evaluation of the patients is done manually by the specialist on the basis of the recorded videos of the session.

On the other hand, the system uses algorithms, taken from previous research [2], to reinforce collected data using facial expression and body language analysis. The endowing of this software architecture within the hardware structure of CLARC is one of the most significant differences of the proposed system with respect to other competitors.

VI. Conclusions and Future work

In this paper we have presented CLARC an autonomous robot to help the clinician in the CGA process. CLARC discharges the clinician from some of the most time consuming CGA procedures, those of performing tests to patients and relatives, and allows him/her to focus on the most important part, the creation of a customized treatment and follow up plan. Currently a fully functional but restricted version of CLARC has been developed, allowing to perform basic *Barthel*, *Mini-Mental* and *Get up and Go* tests. During the next two years the system will be improved to obtain a commercial product, and several other CGA tests will be added.

The Deliberative component will be endowed with more complex execution monitoring features. The scoring process for observation-based tests will be automatically learnt from annotations of medical experts on real sequences using Machine Learning techniques. These techniques will be also used to parametrize the tests, for example learning the questions where patients need more time to answer, or further explanations. Also the whole use case, from patient greeting to good-bye will be encoded in PDDL and executed, reacting and generating new plans when something not expected happens.

[4]http://echord.eu/essential_grid/assesstronic/

[5]http://www.aal-europe.eu/projects/alias/

Acknowledgment

This work has been partially funded by the European Union ECHORD++ project (FP7-ICT-601116) and the TIN2015-65686-C5-1-R Spanish Ministerio de Economía y Competitividad project.

References

[1] Luis Vicente Calderita, Luis J Manso, Pablo Bustos, Cristina Suárez-Mejías, Fernando Fernández, and Antonio Bandera. Therapist: Towards an autonomous socially interactive robot for motor and neurorehabilitation therapies for children. *JMIR Rehabil Assist Technol*, 1(1):e1, 2014.

[2] F Cid, J Moreno, P Bustos, and P Nez. Muecas: A multi-sensor robotic head for affective human robot interaction and imitation. *Sensors*, 14(5):7711–7737, 2014.

[3] Carol Dematteo and Nancy Pollock. Quality of Upper Extremity Skills Test. *Physical Occupational Therapy in Pediatrics*, 13:1–18, 1992.

[4] M. Folstein, S. Folstein, and P. McHugh. Mini-mental state: a practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research*, 12:189–198, 1975.

[5] José Carlos González, José Carlos Pulido, Fernando Fernández, and Cristina Suárez-Mejías. Planning, Execution and Monitoring of Physical Rehabilitation Therapies with a Robotic Architecture. In Ronald Cornet, Lúacrúamioara Stoicu-Tivadar, Alexander Hörbst, Carlos Luis Parra Calderón, Stig Kjær Andersen, and Mira Hercigonja-Szekeres, editors, *Proceedings of the 26th Medical Informatics Europe conference (MIE)*, volume 210 of *Studies in Health Technology and Informatics*, pages 339–343, Madrid (Spain), May 2015. European Federation for Medical Informatics (EFMI), IOS Press.

[6] LC Gray and R. Wootton. Comprehensive geriatric assessment ”online”. *Australasian Journal on Ageing*, 27(4):205–208, 2008.

[7] A Guneysu, RD Siyli, and AA Salah. Auto-evaluation of motion imitation in a child-robot imitation game for upper arm rehabilitation. *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pages 199–204, 2014.

[8] César Guzmán, Vidal Alcázar, David Prior, Eva Onaindía, Daniel Borrajo, Juan Fdez-Olivares, and Ezequiel Quintero. Pelea: a domain-independent architecture for planning, execution and learning. In *Proceedings of ICAPS'12 Scheduling and Planning Applications woRKshop (SPARK)*, pages 38–45, Atibaia (Brazil), 2012. AAAI Press.

[9] F.I. Mahoney and D. Barthel. Functional evaluation: the Barthel index. *Maryland State Med Journal*, 14:56–61, 1965.

[10] Alejandro Martín, José C. González, José C. Pulido, Ángel García-Olaya, Fernando Fernández, and Cristina Suárez. Therapy monitoring and patient evaluation with social robots. In *Proceedings of the 3rd 2015 Workshop on ICTs for Improving Patients Rehabilitation Research Techniques*, REHAB '15, pages 152–155, New York, NY, USA, 2015. ACM.

[11] S. Mathias, USL Nayak, and B. Isaacs. Balance in elderly patients: the ”get-up and go” test. *Arch Phys Med Rehabil*, 67:387–389, 1986.

[12] Drew McDermott. PDDL - the planning domain definition language. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.

[13] W Tao, T Liu, R Zheng, and H Feng. Gait analysis using wearable sensors. *Sensors (Basel, Switzerland)*, 12(2), 2012.