SRS

Multi-Role Shadow Robotic System for Independent Living

Small or medium scale focused research project (STREP)

DELIVERABLE **D1.3**

SRS Initial Knowledge-base and SRS hardware, software communication, and intelligence specification

Contract number :	247772
Project acronym :	SRS
Project title :	Multi-Role Shadow Robotic System for Independent Living

Deliverable number :	**D1.3**
Nature :	R – Report
Dissemination level :	PU – PUBLIC
Delivery date :	06-Sep-2010

Author(s) :	Dr. Renxi Qiu
Partners contributed :	All SRS members
Contact :	Dr Renxi Qiu, MEC, Cardiff School of Engineering, Cardiff University, Queen's Buildings, Newport Road, Cardiff CF24 3AA, United Kingdom Tel: +44(0)29 20875915; Fax: +44(0)29 20874880; Email: QiuR@cf.ac.uk

The SRS project was funded by the European Commission under the 7th Framework Programme (FP7) – Challenges 7: Independent living, inclusion and Governance

Coordinator: Cardiff University

Table of Contents

1 SRS Software, Communication and Intelligent Specification

1.1 SRS Architecture Requirement

SRS focuses on the development and prototyping of remotely-controlled, semi-autonomous robotic solutions in domestic environments to support elderly people. It involves significant amount of software development to achieve the targeted outcome. The following requirements on the software architecture have been identified based on the user requirement study and the technology assessment carried out in task 1.1 and task 1.4 of WP1:

Robotic Hardware Independence: As the project is targeted on converting various existing robotic solutions into semi-autonomous robotic carers, the software must be as robot hardware independent as possible. To archive this target, some software modules in the architecture need function as device drivers and thus are tied to hardware. The rest of modules should operate only on higher level hardware-independent abstractions. In the SRS project, the robotic hardware independence will be tested on the two prototypes in the SRS which have completely different hardware.

Parallel Processing: Applications involved in SRS requires considerable amount of computational resources for planning and control to sustain the local intelligence of the robot. At the same time, the SRS software has to satisfy long term learning and analysis requirement. Furthermore, it should also satisfy real-time constraints required by the real world applications. Generally, the onboard computational resources of the robot cannot support all the required computation, so separation the computational load across multiple sources e.g. off board machines is required.

Modularity and Collaborative Development: SRS project involves dozens of researchers and developers from different organisation, discipline and background. Since it targets to build large systems contributing to a sizable code base, it is of high importance to enforce modularity and interoperability between the software components and organise them in a systematic way allowing concurrent work on the system from all the partners in a fashion that components can be developed and verified separately and integrated efficiently in the future.

Cross Platform Communication: SRS communication requires transfer of data and commands between various hardware platforms, operating systems and applications. In order to achieve a versatile concept of robot communications it is very useful to build the SRS communications based on an efficient and reliable foundation. Furthermore, although most of robotic resources are available within Linux environment, some sensors and development kit come with only binary Windows drivers. Therefore, SRS software system must be able to deal with multiple operating system, and cross platform communication is required.

Integration with other Code Base: SRS intended to take the advantage of the latest progress of robotic development. It should be capable of re-use code available from other source. For example identified suitable candidates are the navigation system, and simulators from the Player project, vision algorithms from OpenCV, and planning algorithms from OpenRAVE, among many others. In each case, it should only to expose various configuration options and to route data into and out of the respective software, with as little wrapping or patching as possible.

To satisfy the above requirement, the following frameworks have been identified as potential candidate to form the foundation of the SRS.

<u>Candidate Robotic Frameworks for SRS project</u>

OpenJAUS (Open Joint Architecture for Unmanned Systems)
Purpose: Support the acquisition of Unmanned Systems by providing a mechanism for reducing system life-cycle costs. This is accomplished by providing a framework for technology reuse/insertion.
Technical constraints:

- Platform Independence
- Mission Isolation
- Computer Hardware Independence
- Technology Independence

ORCA
Purpose: Orca is an open-source suite of tools for developing component-based robotic systems. It provides the means for defining and developing components which can be pieced together to form arbitrarily complex robotic systems, from single vehicles to distributed sensor networks. In addition it provides a repository of pre-made components which can be used to quickly assemble a working robotic system
Technical constraints:

- Little flexibility with regard to the implementation platform

OROCOS (Open Robot Control Software)
Purpose: The Open Robot Control Software project provides a Free Software toolkit for real-time robot arm and machine tool control. Consists of two decoupled but integrated sub-projects:

- Open Real-time Control Services.
- Open Robot Control Software.

Technical constraints:

- The Orocos project seems to contain fine C++ libraries useful for industrial robotic applications and is focused on control software

ROS (Robot Operating System – Robot Open Source)
Purpose: ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device

control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers

Technical constraints:

- No technique limitations can be found

PLAYER

Purpose: The Player Project creates Free Software that enables research in robot and sensor systems. According to the Player Project, the Player robot server is probably the most widely used robot control interface in the world. Its simulation back-ends, Stage and Gazebo, are also very widely used. Released under the GNU General Public License, all code from the Player/Stage project is free to use, distribute and modify. Player is developed by an international team of robotics researchers and used at labs around the world.

Technical constraints:

- It is mostly US funded by NSF, DARPA and JPL and supported by US research institutions
- Recent development is relatively slow

MICROSOFT ROBOTICS

Purpose: According to Microsoft, Microsoft Robotics products and services enable academic, hobbyist and commercial developers to easily create robotics applications across a wide variety of hardware.

Technical constraints:

- Dependency on Microsoft development tools
- Limited hardware support

CLARAty (Coupled-Layer Architecture for Robotic Autonomy)

Purpose: CLARAty is an integrated framework for reusable robotic software. It defines interfaces for common robotic functionality and integrates multiple implementations of any given functionality. Examples of such capabilities include pose estimation, navigation, locomotion and planning. In addition to supporting multiple algorithms, CLARAty provides adaptations to multiple robotic platforms. CLARAty, which was primarily funded by the Mars Technology Program, serves as the integration environment for the program's rover technology developments.

Technical constraints:

- Public access seems to be limited.
- The license and download policy has critics.
- CLARAty is incompatible with the GPL and cannot be used for commercial activities.

YARP (Yet Another Robot Platform)

Purpose: It is a set of libraries, protocols, and tools to keep modules and devices cleanly decoupled. It is reluctant middleware, with no desire or expectation to be in control of your system. YARP is definitely not an operating system.

Technical constraints:

- Yarp / RoboCub were supported by European Union grant RobotCub (IST- 2004-004370) and by euCognition (FP6 Project 26408). These excellent projects have ended.

CARMEN (Carnegie Mellon Robot Navigation Toolkit)

Purpose: CARMEN is an open-source collection of software for mobile robot control. CARMEN is modular software designed to provide basic navigation primitives including: base and sensor control, logging, obstacle avoidance, localization, path planning, and mapping.

Technical constraints:

- C programming language
- No graphical tools
- Not vision/speech processing

MOOS (Mission Oriented Operating Suite)

Purpose: MOOS is a C++ cross platform middle ware for robotics research. It is helpful to think about it as a set of layers.

- Core MOOS - The Communications Layer: The most fundamental layer CoreMOOS is a very robust network based communications architecture (two libraries and a lightweight communications hub called MOOSDB) which for very little effort lets you build applications which communicate with each other.
- Essential MOOS - Commonly Used Applications: Essential MOOS is a layer of applications which use CoreMOOS. They offer a range of functionality covering common tasks for example process control, logging

Technical constraints:

- Oriented to autonomous marine vehicles

RoboComp

Purpose: RoboComp is an open-source robotic software framework. It uses software component technology to achieve its goals: efficiency, simplicity and reusability. Its components can be distributed over several cores and CPU's. Existing software components, can be easily integrated with new components made by RoboComp users.

Technical constraints:

- Rough list of common software dependences
- Communication depends on the ICE framework
- Still under development

MARIE

Purpose: MARIE is a free software tool using a component based approach to build robotics software systems by integrating previously existing and new software components.

MARIE's initiative is based on the following main requirements:

- Reuse softwares, APIs, middlewares and frameworks frequently used in robotics (Player, CARMEN, RobotFlow, etc.)
- Adopt a rapid-prototyping approach to build complete system
- Allow distributed computing on heterogeneous platforms
- Allow concurrent use of different communication protocols, mechanisms and standards
- Accelerate user-defined developments with well defined layers, interfaces, frameworks and plugins
- Support multiple sets of concepts and abstractions

Technical constraints:

- Low level communications partially supported
- No security provided
- Incomplete documentation

Design Choices

To meet the requirements, after careful consideration we have identified ROS as the best candidate reference architecture for the SRS project. It satisfies all the requirement listed in the sections above. In more detail, ROS supports parallel processing through message passing along a user-defined, task-specific graph of connections between software modules. Modularity is enforced through the operating system process model: each software module executes as a process on some CPU. The TCP protocol was chosen for message passing, because it is supported on all modern operating systems and networking hardware. Its operation is essentially lossless. It provides standard operating system services such as hardware abstraction, low-level device control to ensure robot independent and has good support for sensors and various development kits. The ROS peer-to-peer links communicate with each other by passing messages also helps the Cross Platform Communication. Finally it has good support for code-reuse.

1.2 SRS Framework Components

In this session, SRS framework components will be specified in the following steps:

(1) Definition of SRS tasks and actions based on Scenarios identified in task 1.1 of WP1;
(2) Identification of the components required in the tasks/actions execution ;
(3) Analysis of the expected interaction among the identified components.

Following the above steps a certain number of task and their subtasks have been derived from the SRS scenarios. They are listed in the table 1 and table 2 below. In each table the tasks are divided into sub-tasks. For each sub-task, an appropriate action is identified with pre-condition and post-condition for their successful execution (see the table). Suitable component has been identified so that it can execute the sub-task. In such a way a complete list of the components involved in the SRS Framework has been derived and it is listed in the Figure 1. As it can be seen from the table a single component is used to execute many subtasks. This requirement is aimed to increase the reusability of the components in the SRS framework.
The interactions among the SRS components are listed in more detail in the table 3.

Task Vs Components for SRS Prototype I Scenarios (Table 1)

SCENARIO - TASKS	SUB-TASKS	ACTIONS	COMPONENT (Decision making communicates with ...)	PRE-CONDITIONS	POST-CONDITIONS
Scenario 1 – Bring and object located on a table	Get Order	Get Starting Signal	Local/ Remote User Interface		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Find Table	Move base to scan position	Navigation	Target position specified	Position reached
		Build 3D map	Environment Perception		3D map generated
		Locate table	Environment Perception	3D map available	Table extracted
	Move base to table	Move base to table	Navigation	Target position specified	Position reached
	Find object	Locate object	Object Detection	Object is in knowledge base Robot is in “find-object” position	Object recognized
	Grasp object	Update 3D map	Environment Perception		3D map updated
		Move arm to pre-grasp position	Manipulation	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Move gripper to open position	Manipulation	Pre-grasp position reached Grasp configuration available	Gripper open
		Move arm to grasp position	Manipulation	Gripper is open grasp position reachable grasp position specified	grasp position reached
		Move gripper to close position	Manipulation	Gripper is open grasp position reached Grasp configuration available	Object grasped
	Place object on tray	Move tray to up position	Manipulation		Tray is up
		Move arm to tray position	Manipulation	tray position specified	Tray position reached
		Move gripper to open position	Manipulation	Tray position reached Tray is up	Gripper is open
		Move arm to folded position	Manipulation	Gripper is empty	Folded position reached
		Move gripper to close position	Manipulation	Gripper is empty	Gripper is closed
SCENARIO - TASKS	SUB-TASKS	ACTIONS	COMPONENT (Decision making communicates with ...)	PRE-CONDITIONS	POST-CONDITIONS

Scenario 1- Setting table	Get Order	Get Starting Signal	Local/ Remote User Interface		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Find Shelf	Move base to scan position	Navigation	Target position specified	Position reached
		Build 3D map	Environment Perception		3D map generated
		Locate shelf	Environment Perception	3D map available	Shelf extracted
	Move base to shelf	Move base to shelf	Navigation	Target position specified	Position reached
	Open shelf	Locate handle	Object Detection Environment Perception	Object is in knowledge base Robot is in “find-object” position	Object recognized
		Move arm to pre-grasp position	Manipulation	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Move gripper to open position	Manipulation	Pre-grasp position reached Grasp configuration available	Gripper open
		Move arm to grasp position	Manipulation	Gripper is open grasp position reachable grasp position specified	grasp position reached
		Move gripper to close position	Manipulation	Gripper is open grasp position reached Grasp configuration available	Object grasped
		Move arm and base synchronously to open door	Manipulation Navigation	Door open trajectory available/possible	Door open position reached
		Move gripper to open position	Manipulation	Door open position reached	Gripper is open
		Move arm to folded position	Manipulation	Gripper is empty	Folded position reached
		Move gripper to close position	Manipulation	Gripper is empty	Gripper is closed
	Move base to shelf	Move base to shelf	Navigation	Target position specified	Position reached
	Find object	Locate object	Object Detection	Object is in knowledge base Robot is in “find-object” position	Object recognized
	Grasp object	Update 3D map	Environment Perception		3D map updated
		Move arm to pre-grasp position	Manipulation	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Move gripper to open position	Manipulation	Pre-grasp position reached Grasp configuration available	Gripper open

		Move arm to grasp position	Manipulation	Gripper is open grasp position reachable grasp position specified	grasp position reached
		Move gripper to close position	Manipulation	Gripper is open grasp position reached Grasp configuration available	Object grasped
		Move arm to transport position	Manipulation	Transport position specified	Transport position reached
	Find Table	Move base to scan position	Navigation	Target position specified	Position reached
		Build 3D map	Environment Perception		3D map generated
		Locate table	Environment Perception	3D map available	Table extracted
	Move base to table	Move base to table	Navigation	Target position specified	Position reached
	Place object on table	Update 3D map	Environment Perception		3D map updated
		Move arm to delivery position	Manipulation	Delivery position reachable	Delivery position reached
		Move gripper to open position	Manipulation	Delivery position reached	Gripper is open
		Move arm to folded position	Manipulation	Gripper is empty	Folded position reached
		Move gripper to close position	Manipulation	Gripper is empty	Gripper is closed
SCENARIO - TASKS	**SUB-TASKS**	**ACTIONS**	**COMPONENT (Decision making communicates with ...)**	**PRE-CONDITIONS**	**POST-CONDITIONS**
Scenario 1- Heating and serving dinner	Get Order	Get Starting Signal	Local/ Remote User Interface		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Find Fridge	Move base to scan position	Navigation	Target position specified	Position reached
		Build 3D map	Environment Perception		3D map generated
		Locate fridge	Environment Perception	3D map available	Fridge extracted
	Move base to fridge	Move base to fridge	Navigation	Target position specified	Position reached
	Open fridge	Locate handle	Object Detection Environment Perception	Object is in knowledge base Robot is in “find-object” position	Object recognized
		Move arm to pre-grasp position	Manipulation	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Move gripper to open position	Manipulation	Pre-grasp position reached	Gripper open

				Grasp configuration available	
		Move arm to grasp position	Manipulation	Gripper is open grasp position reachable grasp position specified	grasp position reached
		Move gripper to close position	Manipulation	Gripper is open grasp position reached Grasp configuration available	Object grasped
		Move arm and base synchronously to open door	Manipulation Navigation	Door open trajectory available/possible	Door open position reached
		Move gripper to open position	Manipulation	Door open position reached	Gripper is open
		Move arm to folded position	Manipulation	Gripper is empty	Folded position reached
		Move gripper to close position	Manipulation	Gripper is empty	Gripper is closed
	Move base to fridge	Move base to fridge	Navigation	Target position specified	Position reached
	Find object	Locate object	Object Detection	Object is in knowledge base Robot is in “find-object” position	Object recognized
	Grasp object	Update 3D map	Environment Perception		3D map updated
		Move arm to pre-grasp position	Manipulation	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Move gripper to open position	Manipulation	Pre-grasp position reached Grasp configuration available	Gripper open
		Move arm to grasp position	Manipulation	Gripper is open grasp position reachable grasp position specified	grasp position reached
		Move gripper to close position	Manipulation	Gripper is open grasp position reached Grasp configuration available	Object grasped
		Move arm to transport position	Manipulation	Transport position specified	Transport position reached
	Find Microwave	Move base scan position	Navigation	Target position specified	Position reached
		Update 3D map	Environment Perception		3D map generated
		Locate microwave	Environment Perception	3D map available	Microwave extracted
	Move base to microwave	Move base to microwave	Navigation	Target position specified	Position reached
	Open microwave*)	Locate handle	Object Detection Environment Perception	Object is in knowledge base Robot is in “find-object”	Object recognized

				position	
		Move arm to pre-grasp position	Manipulation	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Move gripper to open position	Manipulation	Pre-grasp position reached Grasp configuration available	Gripper open
		Move arm to grasp position	Manipulation	Gripper is open grasp position reachable grasp position specified	grasp position reached
		Move gripper to close position	Manipulation	Gripper is open grasp position reached Grasp configuration available	Object grasped
		Move arm and base synchronously to open door	Manipulation Navigation	Door open trajectory available/possible	Door open position reached
		Move gripper to open position	Manipulation	Door open position reached	Gripper is open
		Move arm to folded position	Manipulation	Gripper is empty	Folded position reached
		Move gripper to close position	Manipulation	Gripper is empty	Gripper is closed
	Place object in microwave	Update 3D map	Environment Perception		3D map updated
		Move arm to delivery position	Manipulation	Delivery position reachable	Delivery position reached
		Move gripper to open position	Manipulation	Delivery position reached	Gripper is open
		Move arm to folded position	Manipulation	Gripper is empty	Folded position reached
		Move gripper to close position	Manipulation	Gripper is empty	Gripper is closed
	Close Microwave	Move base to door open position	Navigation	Door open position specified	Door open position reached
		Move arm and base synchronously to close door	Manipulation Navigation	Door closed trajectory available/possible Position of door/ handle stored	Door closed position reached
	Activate Microwave	Locate button	Object Detection Environment Perception	Object is in knowledge base Robot is in “find-object” position	Button detected
		Move arm to press button position	Manipulation	Button position specified Button reachable	Button pressed
	Open Microwave	See above			
	Find Object	Optional, see above			
	Grasp Object	See above			
	Find table	See above			
	Move base to table	See above			

	Place object on table	See above			
SCENARIO -TASKS	SUB-TASKS	ACTIONS	COMPONENT (Decision making communicates with ...)	PRE-CONDITIONS	POST-CONDITIONS
Scenario 1 – Night monitoring II- SRS in a intelligent Environment	Locate fall of person	Locate fall of person	Intelligent Home		Position of person available (Room number)
	Get Order	Get Activation Signal	Intelligent Home	Position of person available (Room number)	
	Move base to room	Move base to room	Navigation	Position of person available (Room number)	Target position reached
	Locate person	Locate person	Human Motion Detection	Target position reached	Person position specified
	Move base to person	Move base to person	Navigation	Person position available	Target position reached
	Enable Communication	Enable Communication	Remote User Interface		Connection established

*) Some microwaves are opened by pressing a button instead

Locate: Perception actions

Move: Movement actions

Get/Enable/Update/Build: Trigger actions

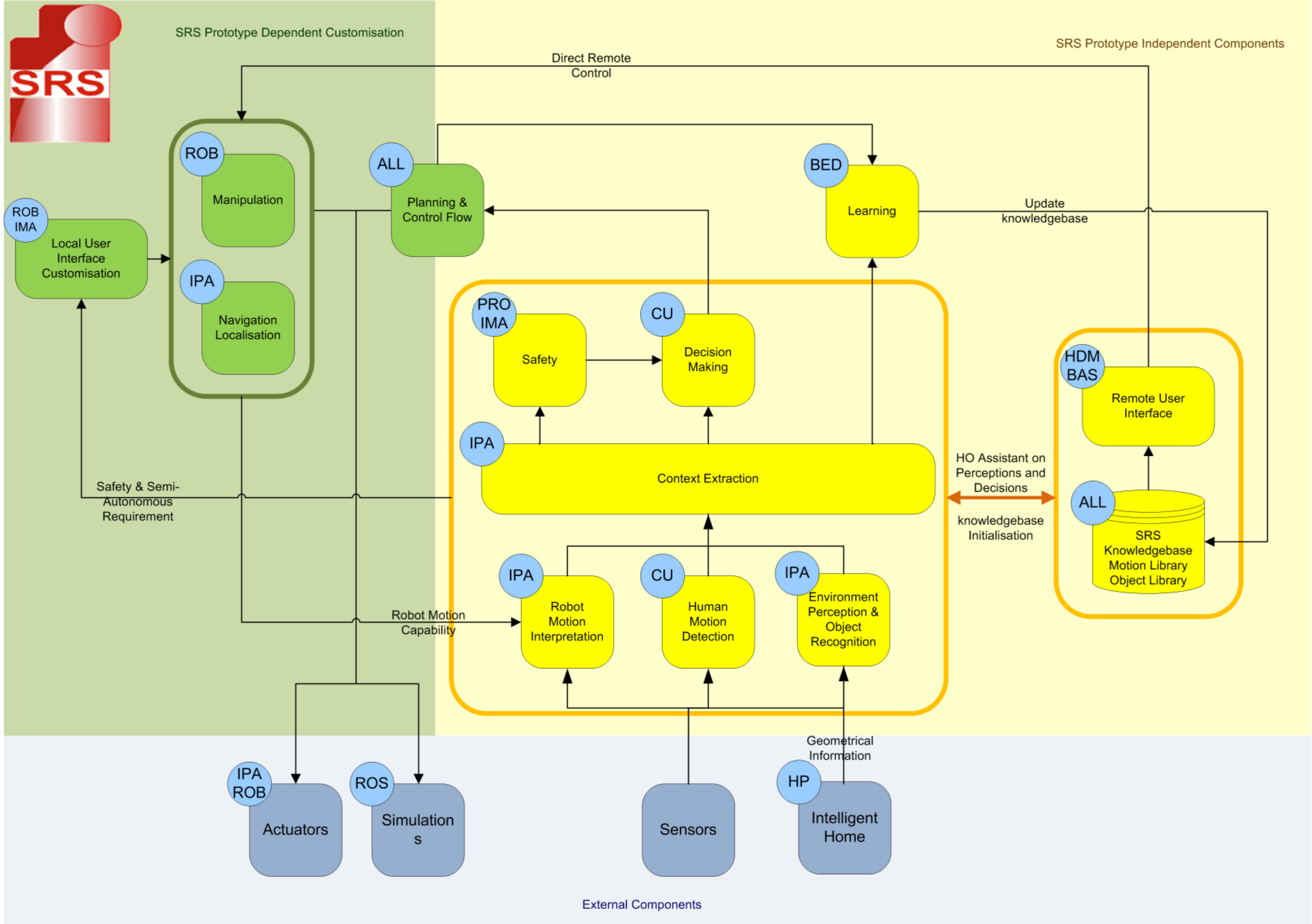
Task Vs Components for SRS Prototype II Scenarios (Table 2)

Scenario TASK	SUB-TASKS	ACTIONS	COMPONENT (decision making communicates with…)	PRE-CONDITIONS	POST-CONDITIONS
Scenario 2 – Day monitoring	Get monitor Order	Get Starting Signal	Remote User Interface		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Move base around	Move Base around	Navigation	Target position specified	Position reached
	Monitor	Update 3D map	Environment perception Monitor module*1	3D Map available	3D map updated
Scenario 2 – Standing up assistance	Get standing up Order	Get Starting Signal	Local/Remote user interface		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Move base to sofa	Move Base to sofa	Navigation	Target position specified	Position reached
Scenario 2 – Fetch a book from a shelf	Get fetch book Order	Get Starting Signal	Local user interface		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Find shelf*2	Move Base to appropriate scan position	Navigation	Target position specified	Position reached
		Build 3D map	Environment Perception		3D map generated
		Extract shelf	Environment Perception	3D map available	Shelf extracted
	Move base to shelf	Move Base to shelf	Navigation	Target position specified	Position reached
	Find book	Find object	Object Detection	Object is in knowledge base Robot is in "find-object" position	Object recognized
	Grasp book	Update 3D map	Environment Perception		3D map updated
		Move Arm to Pre-Grasp position	Manipulation	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Open Gripper	Manipulation	Pre-grasp position reached Grasp configuration available	Gripper open
		Move Arm to grasp position	Manipulation	Gripper is open grasp position reachable	grasp position reached

				grasp position specified	
		Close gripper	Manipulation	Gripper is open grasp position reached Grasp configuration available	Object grasped
	Place the book on the platform table	Move arm to table position	Manipulation	table position specified	Table position reached
		Release Object	Manipulation	Table position reached	Gripper is open
		Move arm to folded position	Manipulation	Gripper is empty	Folded position reached
		Close Gripper	Manipulation	Gripper is empty	Gripper is closed
Scenario 2 – Shopping Reminder function	Get reminder order	Get Starting Signal	HP application Time and date reminder (weekly)		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Shopping	Show list of goods	Shopping module*3/Local User interface	List of goods available at knowledge base	List of goods
		Selection of new goods	Shopping module*3 /Local User interface	List of goods available at knowledge base	New list of goods
		Send order	Shopping module*3	New list of goods	Order Ack
Scenario 2 – Help with heavy Shopping	Get bring shopping order	Get Starting Signal	Local User Interface		Name of task available
		Get Sub-Task List from knowledge base	Knowledge base	Name of task available	List of sub-tasks available
	Move base to entrance door	Move Base to entrance door	Navigation/Remote operator interface	Target position specified	Position reached Table loaded with sopping
	Move base to kitchen	Move Base to kitchen	Navigation/Remote operator interface	Target position specified	Position reached
	Select object from delivery box	select object	Object Detection/Remote operator interface	Object is visible by the remote operator	Object recognized
	Grasp object from delivery box	Update 3D map	Environment Perception		3D map updated
		Move Arm to Pre-Grasp position	Manipulation/Remote operator interface	Pre-grasp position reachable (object position), pre-grasp position specified	Pre-grasp position reached
		Open Gripper	Manipulation/Remote operator interface	Pre-grasp position reached Grasp configuration available	Gripper open
		Move Arm to grasp position	Manipulation/Remote operator interface	Gripper is open grasp position reachable grasp position specified	grasp position reached
		Close gripper	Manipulation/Remote operator interface	Gripper is open grasp position reached	Object grasped

				Grasp configuration available	
	Place object on platform table	Move arm to table position	Manipulation/Remote operator interface	table position specified	Table position reached
		Release Object	Manipulation/Remote operator interface	Table position reached	Gripper is open
		Move arm to folded position	Manipulation/Remote operator interface	Gripper is empty	Folded position reached
		Close Gripper	Manipulation/Remote operator interface	Gripper is empty	Gripper is closed

Figure 1 SRS Components Diagram



Colour coding has been used to indicate whether the module is hardware specific (in green colour) or hardware independent (yellow colour). Also based on the area of expertise of the project partners the individual modules has been assigned to a partner and this has been indicated in the diagram with the small blue circles on the left-hand corner of the module. The arrows between the components/group of components, shown in the diagram, specify the high level conceptual interactions between the components. These have been elaborated in the table below (Table 3) where the required inputs and outputs from other components together with reference to the tasks and work packages, as specified by DOW, where the specific piece of work will be carried out.

Table 3 SRS Components Interaction

Component ID	Component Name	Description	Inputs	Outputs
C1	Environment Perception & Object Recognition	T3.1	**C13 Sensors** Local information about environment **C12 Intelligent Home** Global information about environment **C9 Remote User Interface** Assistant on object interpretation and calibration	**C4 Context Extraction** Information about Environment Model **C9 Remote User Interface** Feedback of the Environment Information **C10 SRS Knowledgebase** Initial object library
C2	Human Motion Detection	T3.2 T4.1	**C13 Sensors** (2D/3D Cams) **C9 Remote User Interface** Motion Input Assistant on motion interpretation (HO) and local motion detection	**C4 Context Extraction** Estimated Raw Poses Mapped trajectory between robot and HO for position control Detected motion from local site **C10 SRS Knowledgebase** Initial human motion library
C3	Robot Motion Interpretation	T3.3	**C13 Sensors** **C9 Remote User Interface** Assistant on robot motion specification **C14 Manipulation** **C15 Navigation** **Localisation** Robot motion capability	**C4 Context Extraction** High level motion description based on reference motion library **C9 Remote User Interface** Visual feedback of robot motion **C10 SRS Knowledgebase:** Initial reference motion library of robot
C4	Context Extraction	T3.4	**C1 Environment Perception & Object Recognition** Environment Model	**C10 SRS Knowledgebase** Definition of the possible robot states Initial State Machine (FSM) model for robot behaviour specification

			C2 Human Motion Detection Estimated Raw Poses Mapped trajectory Detection of motion from local site **C3 Robot Motion Interpretation** High level motion description **C9 Remote User Interface** Assistant on context extraction **C10 SRS Knowledgebase** Reference object library & reference motion Library for recognition	**C5 Decision Making &** **C6 Safety** Identified states for the FSM **C8 Learning** Environment information
C5	Decision Making	T3.6 T3.7 T4.2	**C4 Context Extraction** Identified states for the FSM **C6 Safety** Cognitive Overload Monitoring Safety oriented motion control **C9 Remote User Interface** User intervention on autonomy level **C10 SRS Knowledgebase** The State Machine (FSM) and its transition rules	**C11 Planning** Switching between: semi-autonomous, fully autonomous and fully remote-controlled. Parameters required for the planning **C10 SRS Knowledgebase:** Initial state transition rules for the FSM
C6	Safety	T2.5 T4.4 T4.5	**C4 Context Extraction** Identified states for the FSM **C9 Remote User Interface** User intervention on safety issues	**C5 Decision Making** Output of cognitive overload monitoring Output of safety oriented motion control **C7 Local User Interface Customisation**
C7	Local User Interface Customisation	T5.3 T5.4	**C5 Safety**	**C14 Manipulation** **C15 Navigation Localisation**
C8	Learning	T3.5 T4.3	**C11 Planning** High level operation information **C4 Context Extraction** Environment information	**C10 SRS Knowledgebase** Updated FSM model, object library and motion library via learning
C9	Remote User	T4.6 T5.1	**C1 Environment**	**C1 Environment Perception &**

	Interface	WP2	**Perception & Object Recognition** Feedback of the Environment Information **C3 Robot Motion Interpretation** Visual feedback of robot motion **C10 SRS Knowledgebase** Formation of SRS knowledge	**Object Recognition** Assistant on object interpretation and calibration **C2 Human Motion Detection** Motion Input, assistant on motion interpretation (HO) and local motion detection **C3 Robot Motion Interpretation** Assistant on robot motion specification **C4 Context Extraction** Assistant on context extraction **C5 Decision Making** User intervention on autonomy level **C6 Safety** User intervention on safety issues
C10	SRS Knowledgebase	Knowledge based for entire project. Motion Library, Object Library, FSM and Rules Linked to various components	**C1 Environment Perception & Object Recognition** Initial object library **C2 Human Motion Detection** Initial human motion library **C3 Robot Motion Interpretation** Initial reference motion library of robot **C4 Context Extraction** Definition of the possible robot states Initial State Machine (FSM) model for robot behaviour specification **C5 Decision Making** Initial state transition rules inside the FSM **C8 Learning** knowledgebase update via learning	**C1 Environment Perception & Object Recognition** Reference object library for recognition **C2 Human Motion Detection** Human motion library **C3 Robot Motion Interpretation** Reference motion library of robot **C5 Decision Making** The FSM state transition rules **C10 Remote User Interface** Formation of SRS knowledge
C11	Planning	Part of T5.2, T5.3 and T5.4	**C5 Decision Making** Control flow and all necessary parameters required for motion planning	**SRS Prototypes (Actuators and simulation)** **C8 Learning** High level operation information
C12	Intelligent Home	T4.7		**C1 Environment Perception & Object Recognition**

				Information about environment
C13	Sensors			**C1 Environment Perception & Object Recognition** **C2 Human Motion Detection** **C3 Robot Motion Interpretation**
C14	Manipulation	Part of T5.2, T5.3 and T5.4	**C7 Local User Interface Customisation**	**SRS Prototypes (Actuators and simulation)** **C3 Robot Motion Interpretation** Robot motion capability
C15	Navigation Localization	Part of T5.2, T5.3 and T5.4	**C7 Local User Interface Customisation**	**SRS Prototypes (Actuators and simulation)** **C3 Robot Motion Interpretation** Robot motion capability

1.3 SRS Intelligent Requirement

SRS intelligent will be implemented in the SRS decision making, learning and knowledgebase components. Their interaction with rest of the framework is extracted from table above and re-listed below:

Component ID	Component Name	Description	Inputs	Outputs
C5	Decision Making	T3.6 T3.7 T4.2	**C4 Context Extraction** Identified states for the FSM **C6 Safety** Cognitive Overload Monitoring Safety oriented motion control **C9 Remote User Interface** User intervention on autonomy level **C10 SRS Knowledgebase** The State Machine (FSM) and its transition rules	**C11 Planning** Switching between: semi-autonomous, fully autonomous and fully remote-controlled. Parameters required for the planning **C10 SRS Knowledgebase:** Initial state transition rules for the FSM
C8	Learning	T3.5 T4.3	**C11 Planning** High level operation information **C4 Context Extraction** Environment information	**C10 SRS Knowledgebase** Updated FSM model, object library and motion library via learning
C10	SRS Knowledgebase	Knowledge based for entire project.	**C1 Environment Perception & Object Recognition** Initial object library	**C1 Environment Perception & Object Recognition** Reference object library for recognition

		Motion Library, Object Library, FSM and Rules Linked to various components	**C2 Human Motion Detection** Initial human motion library **C3 Robot Motion Interpretation** Initial reference motion library of robot **C4 Context Extraction** Definition of the possible robot states Initial State Machine (FSM) model for robot behaviour specification **C5 Decision Making** Initial state transition rules inside the FSM **C8 Learning** knowledgebase update via learning	**C2 Human Motion Detection** Human motion library **C3 Robot Motion Interpretation** Reference motion library of robot **C5 Decision Making** The FSM state transition rules **C10 Remote User Interface** Formation of SRS knowledge

1.3.1 Consciousness

An SRS robot will be able to recognise

- Local user in different postures
- Different types of furniture such as table, cupboard and door, and
- Objects such as bottles, cups and door handle, when approaching to them.

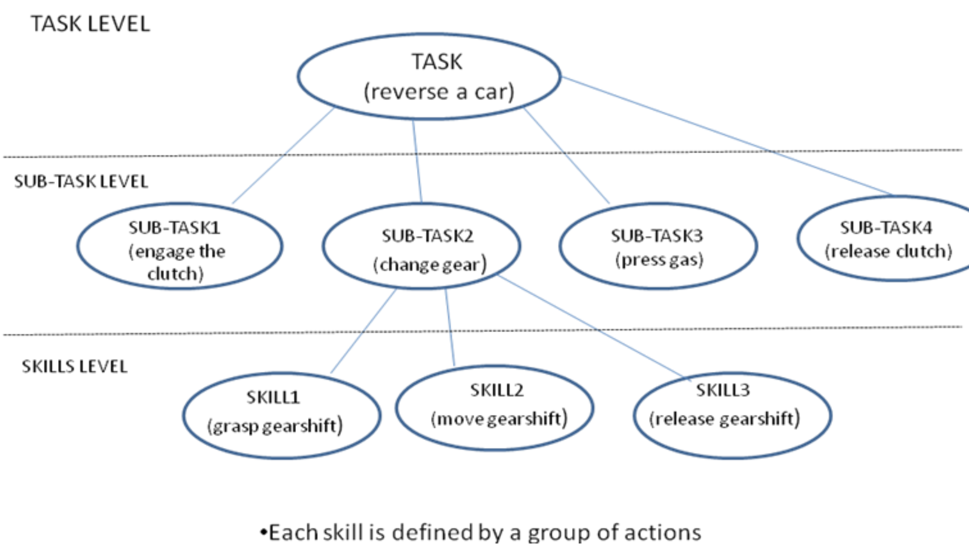
Note: The postures, furniture and objects should be further defined according to testing scenarios used in the project.

1.3.2 User intention recognition

While being manipulated by a remote user, an SRS robot will be able to

- Segment actions it is controlled to perform into sub-tasks
- Identify sub-goals that are associated with the sub-tasks, and
- Recognise the operator's intention through the process of being controlled in completion of a serious of sub-tasks.

-the robot needs at least to have from the beginning the map of the environment (pre-knowledge)
-The "learning mode" starts when the robot leaves the control to the Remote Operator
-All the information provided by the robot has to be recorded for the learning module
-For representing an action, BED will study the possibility of using the "actionlib" package from ROS.
-Actions are needed to represent skills, which will be the information that the Remote Operator will have for the learning module. An example of the hierarchy between skills, sub-tasks, tasks and actions is the next:



1.4 SRS Interaction Technology

1.4.1 Specification procedure

Based on the technology assessment in task 1.3 of WP1 candidate interaction technology for the two main interfaces of SRS, the interface to the local user and the interface to the remote operator, have been selected. In order to take the final decision on the interaction technology it is necessary to analyse which type of interaction technology is best suited for the expected interaction tasks and requirements.

The specification procedure has the following steps:

(1) Define the SRS user interfaces
(2) Select candidate interaction technology
(3) Analyse the expected interaction tasks and requirements
(4) Assess the candidate interaction technology on the basis of tasks and requirements
(5) Take the final decision on interaction technology in task 2.3

1.4.2 SRS user interfaces

The original plan of SRS is, that the local user interface is based on the interaction technology the care-o-bot 3 (COB) provides. There are some important reasons why this might be not enough. Regarding the role of the elderly person within the SRS concept, it might be necessary to consider the elderly person another “remote” user and equip the elderly person with a remote operation device for the following reasons:

- SRS prototype 2 and other robots may lack specific interaction capabilities like a touch screen or speech communication that COB has. SRS should be largely independent of a specific robotic platform and the local interface is out of the scope of SRS. However, interaction with the elderly person cannot be avoided (elderly person initiates the interaction, has the best knowledge of the own apartment, e.g. location of objects, is the only one perfectly informed about her requirements). Therefore local user interaction should take place via a dedicated device designed as part of the SRS UI concept. The device can be largely similar to the device of the relatives but may need to be adapted and reduced in functionality.

- The elderly person will often be in a seated position when interacting with the robot. The COB tray is too high to be operated by a seated person. A different device is therefore necessary.

- Calling the robot from a distant room will be difficult or impossible with the COB local user interface (fixed-position touch screen or speech). Therefore the elderly person needs another device.

- SRS is not targeting elderly persons with severe cognitive impairments but such with none or mild limitations (mainly physical, not mental). Therefore, it is feasible to equip the elderly person with an interaction device. If we can create a highly encouraging user interface, it might even be fun for the elderly person to use the device and to teach the robot new things. This would as a side effect also address the problem of cognitive stimulation.

- There are no privacy issues if the elderly person operates the robot

- There will have to be some communication with the remote operator (e.g. video call).

Two other aspects are important in order to define the SRS interface:

(1) SRS should not develop too many interaction concepts for different user interfaces. Ideally, it should be just one for all targeted user groups because in this case all efforts can be focused to achieve the highest quality UI. This user interface could be scalable between the different user groups.

(2) SRS will require both, low-level (e.g. direct control of arm, gripper, navigation) and high-level control (e.g. selecting a room for navigation, pressing buttons for teaching and modifying behaviour, dragging behavioural components around, entering labels for actions) but the focus should clearly be high-level control because that is the area where SRS's innovation will take place (learning, semi-autonomous mode). In the long run, SRS is aiming to make low-level control obsolete. Further, it is important to recall that the DoW states that the goal is to avoid trajectory-copying interaction approaches because of their problems under real-world network conditions (latencies, weak reliability, can lead to instable positions during manipulation, …).

1.4.3 Candidate interaction technology

The following candidates of interaction technology are compared. The short names like "Kin" stand for the interaction concept behind them (i.e., the column "Kin" does not represent Kinect specifically but devices working by that principle, i.e. using a TV and gesture recognition without controller, etc.).

- **"PC":** Standard Windows PC + 1920x1080 24" LCD + mouse + keyboard + webcam with microphone
- **"Fal":** Haptic 3-DOF force feedback controller (e.g. Novint Falcon) + Windows PC + 1920x1080 24" LCD + mouse + keyboard + webcam with microphone
- **"3dm":** 3D mouse (e.g. 3dconnexion) + Windows PC + 1920x1080 24" LCD + keyboard + webcam with microphone

- **"Kin":** Controller-free 3D gesture recognition (e.g. Microsoft Kinect) + Windows PC + FullHD television (min. 40") + speech recognition and output
- **"Six":** Wireless 3D motion-tracking handheld controller (e.g. Wii Remote with accelerometer and optical sensors or state-of-the art Sixense TrueMotion 3D, with magnetic tracking) + FullHD television (min. 40") + speech recognition and output + video camera + microphone
- **"Tab":** Multi-touch tablet computer with state-of-the-art sensors (e.g. Apple iPad) and video camera
- **"Sma":** Modern smartphone with state-of-the-art sensors like accelerometer, compass, gyroscope and video camera (e.g. Apple iPhone 4, Android phones)

1.4.4 Analyse the expected interaction tasks and requirements

This document defines interaction requirements and compares several types of interaction devices for their suitability for SRS operation by the remote operator. The analysis in this document is based on a review of the SRS scenarios, literature review, the SRS ongoing discussion and 13 additional scenarios worked out at HdM. The scenarios were based on various interaction device configurations for remote operation, 9 scenarios were developed from the perspective of the remote operator and 4 from the perspective of the local user. The scenarios were analysed concerning tasks of remote operator and local user related to different usage situations. Further requirements were extracted from the scenario descriptions.

The results of this analysis are used in the next step: "assessment of the candidate interaction technology on the basis of tasks and requirements".

1.4.5 Assessment of the candidate interaction technology on the basis of tasks and requirements

The assessment is done in the following tables.

The following rating is used:
++ meets requirement very well
\+ rather / probably meets requirement
o borderline
\- rather / probably does not meet requirement
-- does not meet requirement at all
n/a not applicable
? unsure
++[1] numbers: see note below table

In the text several remarks are included. The label "remark" indicates them.

The last line "summary" of each table performs a summary based on the current status of the discussion.

1.4.6 General Interaction Device Requirements (Table 3)

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
Well suited for **high-level operation** (GUI-based arrangement, button presses, entering text, assigning objects to classes, pointing at map location, pointing at objects to be manipulated, etc.)	++	++ --[1]	--	--	--	++	o

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
This kind of operation is required for teaching and semi-autonomous operation							
Well suited for **low-level operation without** trajectory copying or low-latency interaction (like smartphone tilting), e.g. assuming Kinect-type interaction only by command gestures	o	o --[1]	--	--	--	o	o
Well suited for **low-level operation with** trajectory copying and low latency interaction (like smartphone tilting) Remark: This is probably not feasible over the Internet.	--	++	++	++	++	+	+
The device (or device combination) is **always on** (important for remote user notifications)	--	--	--	--	--	++	++
All interaction with the device (including in particular the main form of interaction of the device, like trajectory copying in the case of "Kin", "Six", "Fal", "3dm") will probably **work over Internet**, assuming a state-of-the-art home configuration: DSL/cable (16000 kbps downstream, 1000 kbps upstream, ping times around 25 ms) + Wi-Fi 802.11n.	++	--	o	--	--	++	++
The device is suitable for the interaction requirements of **all three SRS user groups** (e.g., call centre needs advanced high-level features, elderly reduced set)	o	-	-	--	--	+	--
Interaction device is **portable** (important for children of elderly)	--	--	--	--	--	+	++
The device and all associated devices are **affordable** (ideally some users have it anyway, and will not have to buy it) Upper row: rating for user group "children" Lower row: Rating for user group "elderly"	++ -	o --	- --	+ --	- --	+ -	+ -
The device combination does not require much **additional space** (important user requirement by elderly)	--	--	--	+	+	++	++
Versatility: Works for remote operation tasks of **many application scenarios** (not only the currently chosen scenarios like preparing food and night monitoring) and for control of **many service robots** (because the SRS concept is independent of a specific robotic platform)	?	?	?	?	?	?	?
Using the device for domestic service robot control is **innovative**	--	o	?	++	o	++[2]	++[2]
Some consortium partners already have **expertise** with the device Remark: HP has implemented control through a touch	++	++	--	o	--	++	?

Requirement	**PC**	**Fal**	**3dm**	**Kin**	**Six**	**Tab**	**Sma**
screen tablet, IPA through Falcon, CU has some first experience with controller-free gestures							
Summary:	**o**	**o**	**-**	**-**	**-**	+	**o**

(1) upper rating if used in combination with conventional computer mouse, lower rating if not
(2) assumes using the latest smartphones and tablet computers in new ways with multi-touch gestures, sensors like accelerometer, gyroscope

1.4.7 Remote Operator Interaction Requirements

Initiate remote session (Table 4)

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
Send request for remote session to LU	++	++	++	++	++	++	++
Accept LU or robot's request for assistance (ideally, the device should be always on in order to receive a request at any time and it should always be with the remote operator)	--	--	--	--	--	+	++
Deny request and optionally specify a later time or forward to a different RO	++	++	++	o[1]	o[1]	++	++
Provide authentication (e.g. password)	++	++	++	o[1]	o[1]	++	++
End remote session	++	++	++	++	++	++	++
Summary:	**+**	**o**	**o**	**o**	**o**	**++**	**++**

(1) rating considers possible difficulties with text entry or speech recognition

Telepresence (Table 5)

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
Video stream of local user's apartment	++	++	++	++	++	++	o
Augmentation of video stream with detected objects and their names: maybe also differentiate between movable objects (e.g. bottle), non-movable objects (e.g. door handle, drawer handle), and persons (e.g. "Mike"). Also, the associated possible actions could be visualized for each object (e.g. door handle: open/close, kitchen work surface: "put object here")	++	++	++	++	++	++	-
Room plan with position of robot and current robot orientation so that the direction of the camera picture can be assessed (this could be achieved by a "torch light" metaphor showing the angle of the camera)	++	++	++	++	++	++	-
Control angle of robot's camera (what does the robot "look at") – if robot supports it (in the case of COB left/right would correlate with robot navigation but up/down would need to be implemented with an additional control)	+	o[1]	o[1]	?	?	+	+
Zoom camera picture: zoom in, zoom out, pan	+	o[1]	o[1]	+	+	++	++
Robot status (battery level, quality of connection, error state like "ready" or "stuck")	++	++	++	++	++	++	++
Current job and current activity (incl. activity history) of robot	++	++	++	++	++	++	-
Optional (to be evaluated if this could be useful): augmentation of video stream with robot's world model (obstacles detected, 3D vision, distance	++	++	++	++	++	++	o

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
between gripper and objects during manipulation, etc.)							
Optional: feedback of manipulator arm position (maybe augmentation on video for simulation of movement before execution)	++	++	++	++	++	++	o
Summary:	++	+	+	+	+	++	-

(1) may require change of controller (from Falcon or 3d mouse to normal mouse)

High-level fetch and carry (Table 6)

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
High-level navigation: specify target location ("go to fridge", "go to living room", e.g. by pointing on a map of the apartment)	++	++	++	++	++	++	+
Zoom map picture: zoom in, zoom out, pan; or: zoom room by pointing at it	+	o[1]	o[1]	+	+	++	++
Point at and select objects in live video stream (e.g. point at detected object for grasping)	+	o[1]	o[1]	?	+	++	++
Place object on tray (COB) or platform (P2)	++	o[1]	o[1]	o[2]	+[2]	++	++
Carry object from specified location A to specified location B (e.g. upper shelf in kitchen on the left of door, dishwasher, person "Mike")	++	++	++	o[2]	+[2]	++	++
Put object (back) to its standard position or another previous position (robot should keep a location list per object of all positions where it was ever fetched in order to facilitate finding it the next time)	++	++	++	o[2]	+[2]	++	++
Search object and bring it to local user (the robot has to go through a location list of a specified object or to scan the apartment to detect the specified object)	++	++	++	o[2]	+[2]	++	++
Summary:	**+**	**o**	**o**	**o**	**+**	**++**	**+**

(1) may require change of controller (from Falcon or 3d mouse to normal mouse)
(2) The ratings assume that this involves some amount of GUI operation (buttons presses, navigation through menu to search for items, etc.)

Low-level control (excluding low-latency and trajectory-copying modes) (Table 7)

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
Low level navigation (similar to driving a car): move forward/backward, left/right, rotate, adjust speed, stop moving Remark: will this be required or is high-level map-based approach sufficient?	o	--	--	--	o	++[1]	o
Manipulator arm control, mode 1: buttons for forward/backward, up/down, left/right, stop (only 1 of the 2 modes may be required)	o	-	-	--	-	o	o

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
Manipulator arm control, mode 2: specify target position in 3D space, stop (only 1 of the 2 modes may be required) Remark: One problem here could be the lack of 3D vision by the operator, so no interaction device combination might be suitable. One approach could be augmentation with distances to surfaces near the gripper during manipulation or near the robot during navigation but this again is independent of the interaction device.	o	+	+	?	+	o	o
Gripper control: open, close, rotate Remark: control of single fingers or degree of gripping also needed?	o	++	o	?	?	+	o
Extend/retract tray; extend/retract arm (these functions may not be needed in the UI and could be done autonomously)	++	++	++	++	++	++	++
Summary:	**o**	**o**	**-**	**--**	**o**	**+**	**o**

(1) HP showed an interesting implementation which we could employ (Weiss et al., 2009)

Teaching (Table 8)

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
View previously taught behaviour (procedures, actions, objects, locations) by category, search function, etc. on all levels of detail (from complete procedures like heating up microwave pasta to fine-grained sub-actions like gripper target positions)	++	+	+	--	--	++	+
Teach a behavioural procedure: either based on a template (e.g. how to clear a dishwasher) or free definition or changing an existing procedure. Specific interactions: adding the procedure, labelling, re-arranging and deleting sub-procedures	++	+	+	--	--	++	o
Test robot procedure execution (newly taught procedures), intervene and adjust during execution	+	+	+	+	+	+	+
Teach new object: small and movable object by low-level grasping, then scan rotating in gripper (COB)	o	+	+	o	+	+	o
Teach new object: label it, assign to class (e.g. “pots”), assign features (e.g. small, medium, large pot), assign locations (e.g. kitchen, upper compartment of leftmost shelf)	++	+	+	--	--	++	o
Teach new object: fixed-position object (e.g. handle of fridge). If robot supports this kind of object detection, identification of an object in a scene without the ability to rotate object in gripper (e.g. camera picture of a refrigerator and remote operator selects the handle or draws a line around the handle)	+	++	++	+	+	+	o

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
Teach new location: specify location of fixed-position object in 2D space on map (X/Y); specify location of small, graspable objects in 3D space (X/Y/Z) or by name of containing object (e.g. "on shelf 4"); (provide new room plan)	+	++	++	+	+	+	o
Edit taught items (procedures, actions, objects, locations): rename, re-arrange between classes, store procedure as a template, copy, paste, delete items, change assigned features to an object, change/delete previously taught locations	++	+	+	--	--	++	o
Communicate new abilities (e.g. taught by other user) to all remote users and important ones also to local user: behavioural procedures (sequence of actions, e.g. lay the table for breakfast), actions (e.g., grasp a bottle), recognizable objects and their classes (e.g. a milk bottle of the brand X, belonging to the class "drink bottles", belonging to "fridge, lower compartment"), known users (e.g. "Mike is a remote user, he has the priority sequence number 2, he is unavailable from 9 to 5pm during the week")	++	++	++	++	++	++	+
Summary:	**++**	**+**	**+**	**--**	**--**	**++**	**o**

Miscellaneous (Table 9)

Requirement	PC	Fal	3dm	Kin	Six	Tab	Sma
Change operation mode (e.g. from low-level arm control to high-level navigation)	++	++	++	++	++	++	++
Emergency-stop high-level and low-level manipulation and navigation (if RO sees that something may brake, spill, etc. robot should return to a safe position or just stop)	++	++	++	++	++	++	++
Receive indication of problems (e.g. "cannot find object", "cannot grasp object")	++	++	++	++	++	++	++
24-hour call centre: work in a typical office	++	+	+	--	--	o	--
24-hour call centre: control several robots, switch between customers	++	++	++	+	+	+	-
Summary:	**++**	**++**	**++**	**-**	**-**	**+**	**-**

1.4.8 Local User Interaction Requirements (Table 10)

Remark: The table in this section contains an additional column “COB” representing the user interface of COB with no alterations (e.g. no addition of speech recognition).

Also, the ratings assume that a local user does NOT keep a PC or TV on all day. Further, they assume that operating COB’s tray in a seated position is not ergonomic for the local user.

Requirement	COB	PC	Fal	3dm	Kin	Six	Tab	Sma
LU initiates job, e.g. to fetch an object now (send request to robot or to RO) Note that robot could be in another room.	--	--	--	--	--	--	++	++
Accept or deny control request by RO (which could come at any moment during the day)	--	--	--	--	--	--	++	++
Receive indication of active remote operation and end of remote operation	+	++	++	++	++	++	++	++
Specify a suitable remote operator to help with a specific task	-	+	+	+	+	+	++	++
Receive notification about robot’s and RO’s current task, status, plans (e.g. “food is ready”) – important notifications should take place through the local interface, e.g. by speech messages	+	+	+	+	+	+	+	+
Tell robot to come so that its local interface can be used by LU (e.g. robot is in kitchen and LU on sofa)	--	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Provide RO or robot with information on needs (e.g. specify shopping list items) – could be done actively by using a “remote” interaction device or passively by telling RO (through local interaction like speech or through “remote” device)	-	+	+	+	+	+	++	++
Video or at least voice communication between RO and LU during interaction: This increases trust (robot controlled by a well-known person), it makes the remote operation easier (e.g. “Grandma, please tell me where you put the asthma spray so I can fetch it with the robot” / “…show me what the object looks like that the robot could not grasp”), and it addresses the loneliness problem. Furthermore, co-operation functionality could influence acceptance and self-esteem of local user and extend autonomous living.	--	o	o	o	+	+	++	++
Receive request from robot or RO to perform an action in the real world (e.g. remove object so robot can pass or manipulate)	-	++	++	++	++	++	++	++

Requirement	**COB**	**PC**	**Fal**	**3dm**	**Kin**	**Six**	**Tab**	**Sma**
Reminder function (if going to be implemented): should support autonomous living by stepwise reminding / indirect cues (e.g. 10 minutes after time for medicine robot just enters the room, 20 min: robot gives cue for taking medication, 30 min: robot asks if it should bring medication or if local user will do it)	+	+	+	+	+	+	+	+
Summary:	**--**	**-**	**-**	**-**	**--**	**--**	**++**	**++**

1.4.9 Conclusions

In the following a conclusion is drawn for the interface technology candidates:

PC

- \+ large screen for displaying a lot of information at the same time
- \+ established and mature graphical user interfaces like windows
- \+ no performance problems
- \+ exact positioning is possible by using the mouse
- – heterogeneous environment with many error sources: video communication problems frequently occur because microphone not on or webcam not connected
- – not mobile (laptop still bulky)
- – has to be booted which takes time and is not convenient compared to always-on devices
- – not suitable as an additional interface for the local user
- – elderly users often have difficulties with using a mouse and the complexity of the OS
- – innovation factor is zero

Conclusion: Versatile. Well-known environment for programmers. Least-risk option. Probably could be used but not innovative, not mobile, many error sources due to varying hardware. Not suitable for elderly (as additional remote operators).

Fal

- \+ haptic feedback; user can "feel" physical boundaries in the real world
- \+ intuitive 3D manipulation
- – operation of a GUI (buttons, etc.) is slow and cumbersome but the SRS interface will be heavily GUI-based due to focus on high-level operation
- – frequent changes between mouse and Falcon are not ergonomic
- – operation of a 7 DOF arm (Care-O-bot) with a 3 DOF interaction device has limitations (according to IPA feedback)
- – keeps resetting to standard position whereas robot arm may still be in another position
- – precision not always sufficient? (reported by some HdM students after an evaluation)
- – arm position during longer periods of operation not ergonomic (elbow needs to rest but cannot; reported by an HdM student evaluation for SRS)
- – Falcon's buttons are not in an ideal position, can be pressed unintentionally when manipulating
- – gripper movements cannot be replicated

Conclusion: Overall, its main strength is low-level manipulation, however without 3D vision it may still be difficult to properly manipulate objects.

3dm

- \+ well suited for low level control for controlling three dimensions at the same time (well proven for virtual reality)
- – has to be learnt, needs some training
- – not mobile
- – has to be booted
- – difficult for standard interactions on 2D interfaces like controlling menus etc.

Conclusion: The disadvantages of the previous two solutions apply here too. Whether haptic Falcon-type interaction or a 3d mouse would be more suitable would have to be evaluated.

Kin

- + novelty factor / has not been done before
- + large screen, HD resolution
- + gestures may be fun to use (however this needs to be verified, especially for longer usage periods and GUI menu operation)
- – gestures could be multitudinous and have to be learned
- – gestures could be initiated unintentionally
- – gestures could be imprecise
- – user may feel "stupid" when continuously using gestures, particularly for menu selection (reported after an HdM student evaluation)
- – selecting letters when typing is cumbersome
- – physically impaired people may not be able to operate it
- – longer periods of operation in a remote operator session will probably be strenuous
- – no portability
- – multiple devices have to be bought (TV, Kinect, sometimes HiFi) and turned on for each RO session – users will not leave them on continuously
- – no recognition of hand for gripping movements

Conclusion: Its strengths are in trajectory-copying low-level operation which will not be suitable for SRS because of Internet latencies; high-level operation of GUIs shows significant drawbacks. There is a substantial amount of risk involved because of high uncertainty with regard to what interaction will work (see question marks in tables in previous chapter). Requests by a remote operator or questions by the robot (e.g. "Please choose the right bottle") can only be received if the TV is on and if the person is standing in front of it. There is also a space problem and several devices have to be bought in order to guarantee seamless operation (TV, controller, camera, PC?). Elderly will probably not be suitable remote operators.

Six

- + large screen, HD resolution
- + higher precision compared to "Kin"
- – medium novelty
- – difficult for selecting from menus (high level control)
- – not mobile
- – has to be booted

Conclusion: This solution shows many analogies to the "Kin" solution. However, there is higher precision, less risk involved (because it has been tried). Still, with regard to teaching the robot procedures and for the operation of GUI's, both solutions show the same disadvantages.

Tab

- + always on (important for notifications or remote operation request)
- + mobile
- + quite innovative if used in new ways (relying heavily on multitouch interaction and making use of smart sensors)
- + excellent for simple multitouch gestures based on a graphical user interface
- + easy to handle also for elderly users because of reality-based interaction and a philosophy of small and simple information appliances
- + some tablet computers have additional input modes like acceleration and orientation sensors (e.g. Apple iPad)
- – still smaller display than desktop screen or TV
- – in case of the Apple iPad there may be hardware access restrictions and the current version does not yet have a video camera (although chances are very high this will be added in 2011 since the current enclosure already shows an empty space for a video camera module and video calling has just been rolled out for the iPhone)

Conclusion: The main advantages are the "always-on" design and high mobility. This fits very well with the SRS scenarios of remote operators helping "wherever they are" (airport, etc.). Screen size compared to a smartphone is still good and should be sufficient. This is the most versatile solution and it could be scaled to all three user groups with basic, standard, and advanced interface versions. There have been studies showing good suitability of touch-based interaction for elderly persons. Overall, it seems this solution has the fewest drawbacks.

Sma

- + highest mobility of all solutions
- + could be used as interface for the local user and the remote operators (relatives, friends etc. but not professional remote operators in a 24h call center)
- + always on (important for notifications or remote operation request)
- + quite innovative
- + excellent for simple multitouch gestures based on a graphical user interface
- + easy to handle also for elderly users because of reality-based interaction and a philosophy of small and simple information appliances
- – very small display
- – in case of the Apple iPhone there may be hardware access restrictions

Conclusion: While there are compelling advantages of a "robot controller in the pocket", it seems very challenging from an interaction design perspective to implement a fully working robot teaching and control solution on the small screen of a smartphone. The small screen introduces severe restrictions. Maybe this could be done in future research and we should focus on first getting a solution to work on a normal screen.

Speech input and output

- + can be used in addition to other input options opening the possibility of deictic references: "take this" combined with a pointing gesture
- + can be mobile as well, but may be not combinable with specific mobile devices
- + can be used when the hands are busy with other tasks
- + depending on the system speech dialogues can be perceived as natural
- + speech commands can be used as shortcuts, if learnt before
- – speech recognition is always a challenge (e.g. optimal positioning of microphone)
- – when using command control, the commands have to be learnt or displayed in a graphical user interface
- – some users are embarrassed to use speech control when other people are around

Conclusion: Should only be considered as an optional additional input/output method.

1.4.10 Final decision on interaction technology in task 2.3

As specified in the DoW the final decision on the interaction technology will be taken in task 2.3 "Definition of interaction technologies and specification of the basic shadow system interaction" based on the analysis presented above.

2 SRS Prototypes Hardware Specification

The hardware of SRS Prototype is largely based on the existing component of SRS partner IPA, IMA and ROBOTNIK. The hardware details are listed in the following sessions.

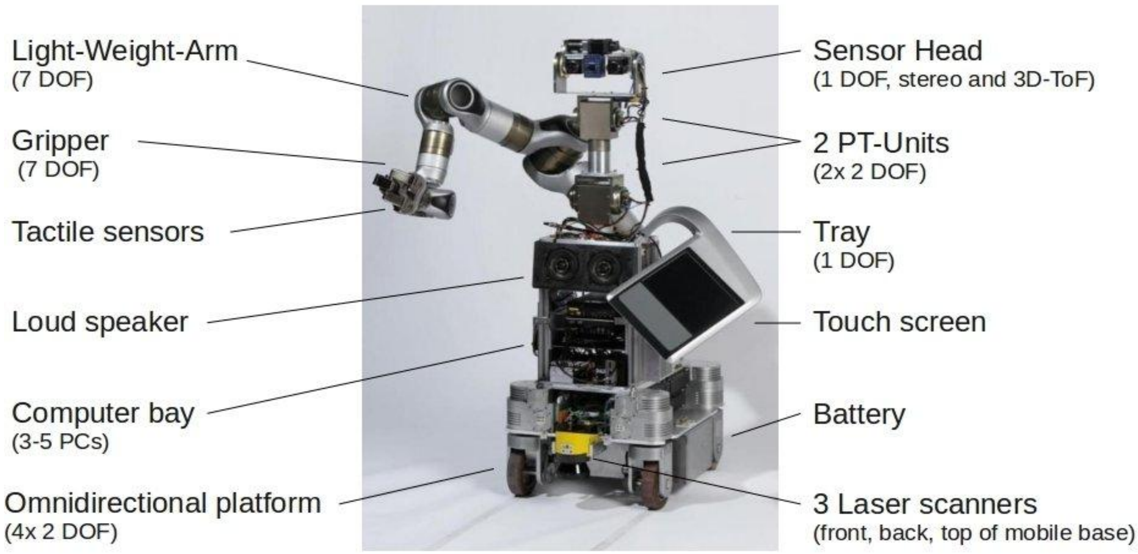
2.1 SRS Prototype I hardware requirements specifications

The prototype I is based on the Care-O-bot® 3 robot from Fraunhofer IPA.

Care-O-bot® 3 Design Concept

- Butler design, not humanoid in order not to raise wrong expectations
- Separation of working and serving side:
  - Functional elements at back, specifically robot arm
  - HRI at front using a tray and integrated touch screen
  - Sensors can be flipped from one side to the other
- Safe object transfer by avoiding contact of user and robot arm
- Intuitive object transfer through tray

Care-O-bot® 3 hardware requirements specifications



Dimensions (L/W/H)	75/55/145 cm
Weight	180 kg
Power supply	Gaia rechargeable Li ion battery 60 Ah, 48 V Internal: 48 V, 12 V, 5 V separate power supplies to motors and controllers All motors connected to emergency-stop circuit
Omnidirectional platform	8 motors (2 motors per wheel: 1 for rotation axis, 1 for drive) Elmo controllers (CAN interface) 2 SICK S300 laser scanners 1 Hokuyu URG-04LX laser scanner Speed: approx. 1.5 m/s
Arm	Schunk LWA 3 (extended to 120 cm) CAN interface (1000 kbaud) Payload: 3 kg

Gripper	Schunk SDH with tactile sensor
	CAN interfaces for tactile sensors and fingers
Torso	1 Schunk PW 90 pan/tilt unit
	1 Schunk PW 70 pan/tilt unit
	1 Nanotec DB42M axis
	Elmo controller (CAN interface)
Sensor head	2 AVT Pike 145 C, 1394b, 1330×1038 (stereo circuit)
	MESA Swissranger 3000/4000
Tray	1 Schunk PRL 100 axis
	LCD display
	Touch screen
Processor architecture	3 PCs (2 GHz Pentium M, 1 GB RAM, 40 GB HDD)

2.2 SRS Prototype II hardware requirements specifications

The prototype II is a robot based on the MOVEMENT platform, the height adjustable table from PROFACTOR/IMA and the modular arm from Robotnik attached with a Barrett Hand:

Hardware specifications of the modular arm are:

- Dimensions: 1100 mm reach
- Weight: 19 Kg
- Payload: 9Kg
- DoF: 6
- 24V Power supply (Batteries needed)
- CAN Bus
- Control PC
- Includes a USB camera for real time vision of grasping tasks

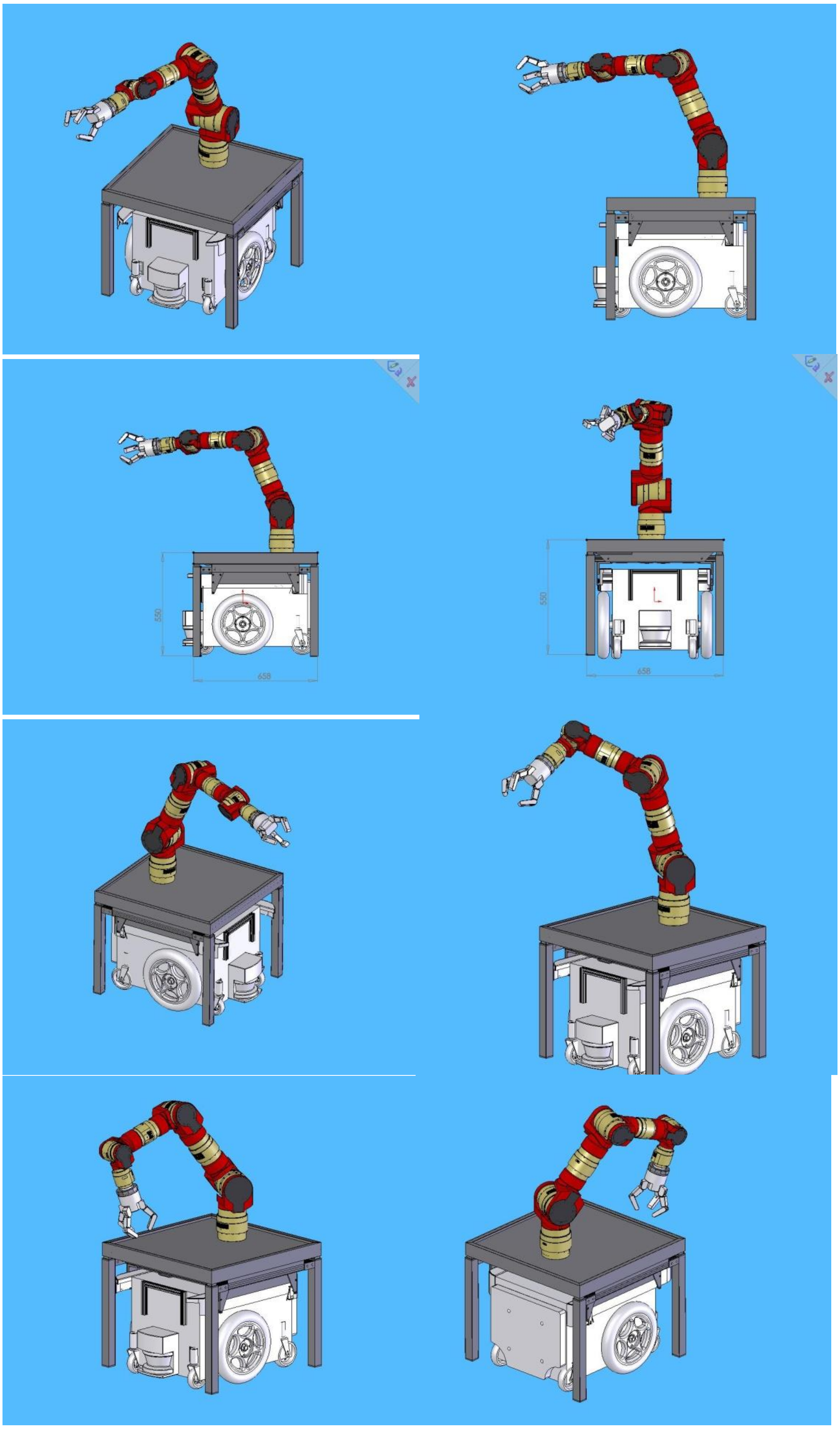
Hardware specifications of the Barrett Hand are:

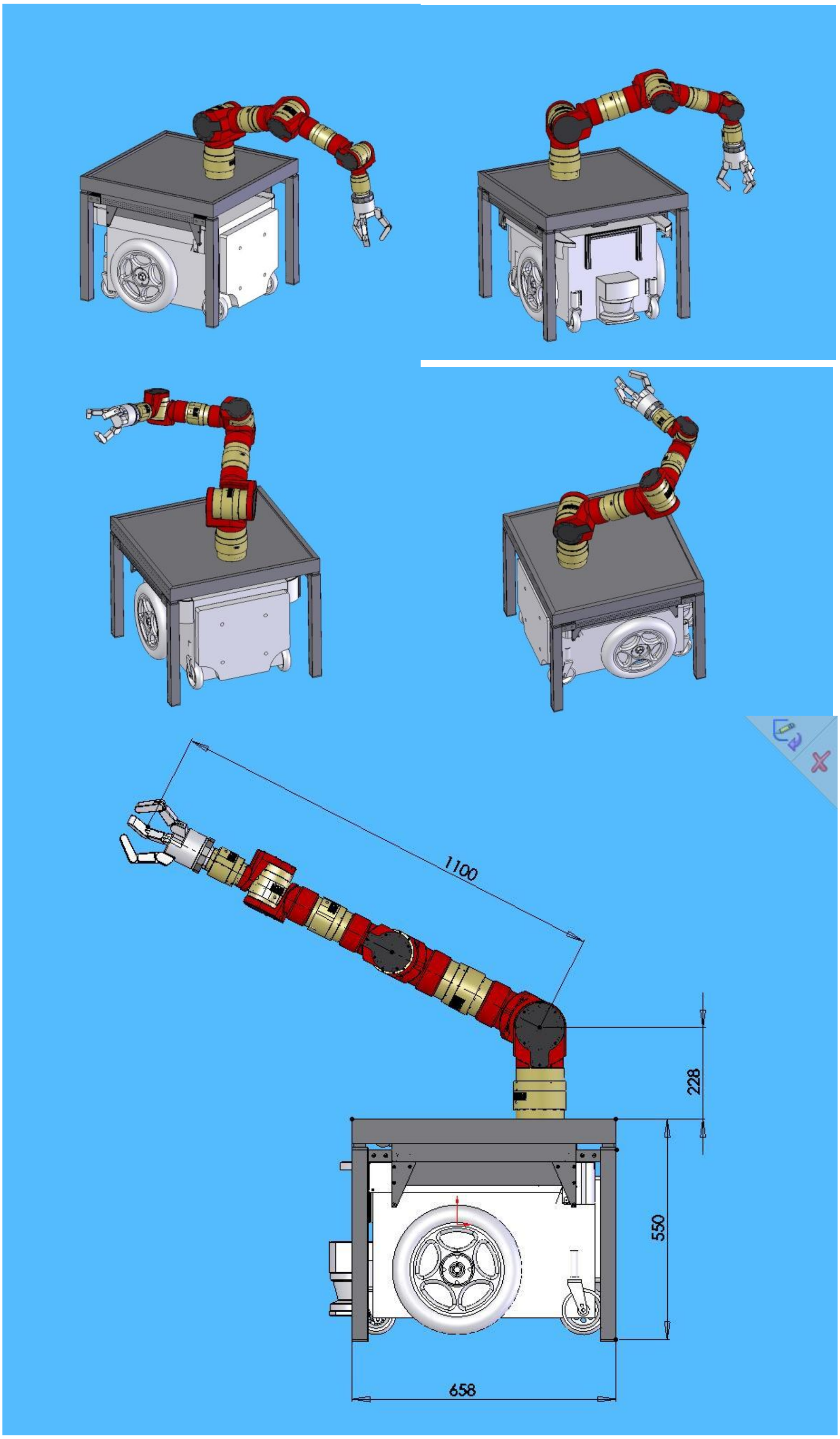
- 5V and 24V power supply
- Serial/CAN Bus
- Payload: 2kg each finger at tip
- Weight: 1.2 Kg
- Shares Control PC with modular arm

The complete system will be upgraded with the next vision modules:

- -Stereo Camera (STOC from Videre)
- -Time of Flight camera

Next figures show a realistic representation of how the complete system will look like (without the vision modules and the handlers for standing assistance):




1100
228
550
658

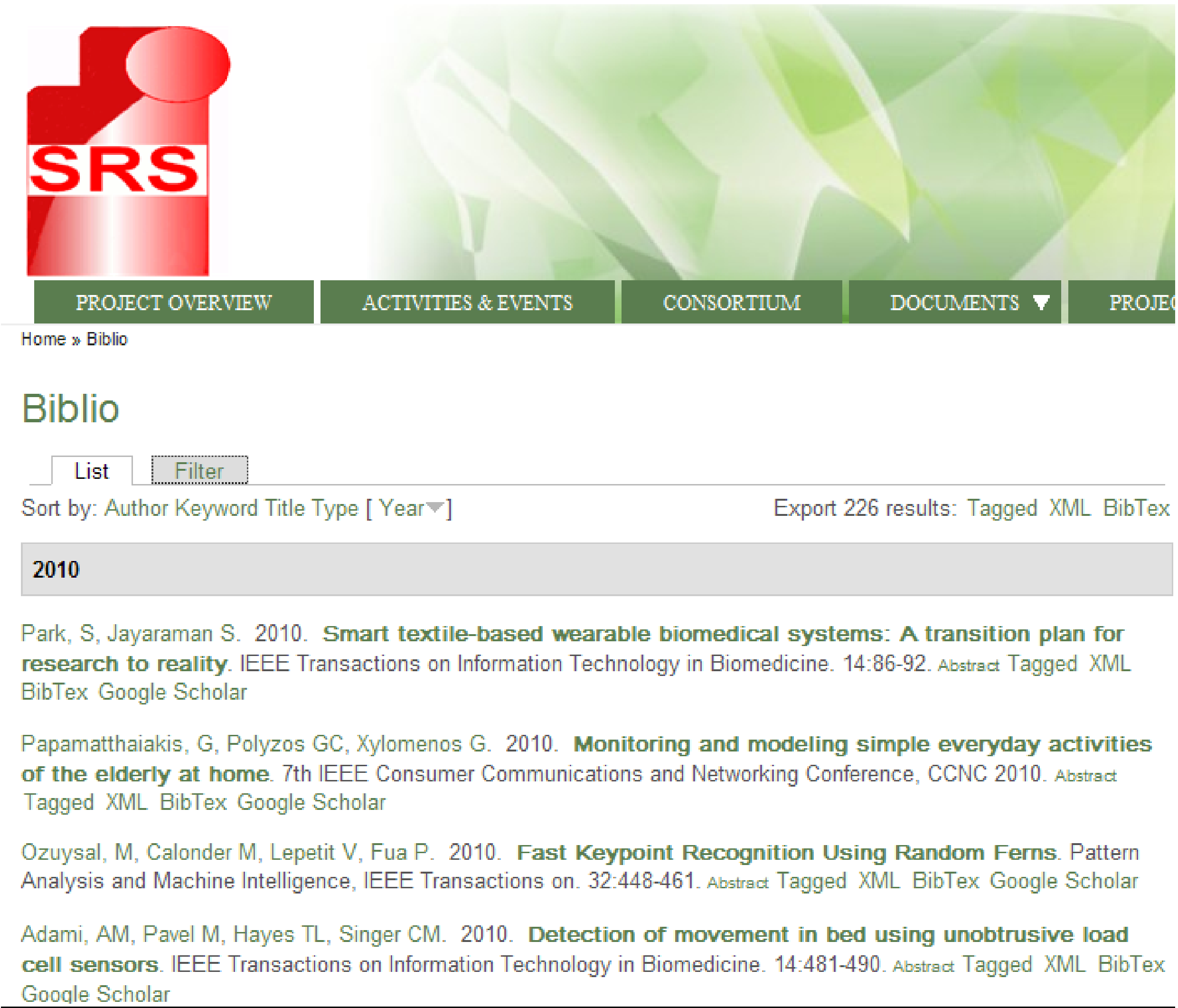
3 SRS Initial Knowledge Base

A database has been developed for clustering detailed technical information on technologies. The database is based on Drupal Scholar. It allows users manage, share and display lists of knowledge through SRS project website.

The database includes the following features:

- Import formats: BibTex, RIS, MARC, EndNote tagged and XML.
- Export formats: BibTex, EndNote tagged and XML.
- Output styles: AMA, APA, Chicago, CSE, IEEE, MLA, Vancouver.
- In-line citing of references.
- Taxonomy integration.

The database has been initialised based on SRS technology assessment. It has 226 records at the moment. Screen shoot of the SRS database is listed below.



Monitoring and modeling simple everyday activities of the elderly at home

View Edit

Title	Monitoring and modeling simple everyday activities of the elderly at home
Publication Type	Conference Proceedings
Year of Conference	2010
Authors	Papamatthaiakis, G, Polyzos GC, Xylomenos G
Conference Name	7th IEEE Consumer Communications and Networking Conference, CCNC 2010
Series Title	2010 7th IEEE Consumer Communications and Networking Conference, CCNC 2010
Conference Location	Las Vegas, NV
ISBN Number	9781424451760 (ISBN)
Abstract	We present our work on a sensor-based smart system automatically trained to recognize the activities of individuals in their home. In this paper we present and analyze a method for recognizing the indoor everyday activities of a monitored individual. This method is based on the data mining technique of association rules and Allen's temporal relations. Our experimental results show that for many (but not all) activities, this method produces a recognition accuracy of nearly 100%, in contrast to other methods based on data mining classifiers. The proposed method is accurate, very flexible and adaptable to a dynamic environment such as the "Smart Home" and we believe that it deserves further attention. Â©2010 IEEE.