Deliverable 6.2 Version 0.0: Manually controlled telepresence system

Project acronym: *TERESA*

Project full title: Telepresence Reinforcement-Learning Social Agent

Grant agreement no.: 611153

Due-Date:	May 2014
Delivery:	M6
Lead Partner:	Giraff (GT)
Dissemination Level:	PU
Status:	Submitted
Version:	Final

DOCUMENT INFO

Date and version number	Author	Comments
2014-06-04 v1.2	Lasse Odens Hedman (GT)	Updated after Jered's review.
2014-05-27 v1.1	Mårten Scherlund (GT)	Added SW section.
2014-05-19 v1.0	Lasse Odens Hedman (GT)	Initial version

Table of Contents

LIST OF TABLES AND FIGURES

1 Executive summary

This deliverable includes the work of assembling all components for the TERESA robot and get it to run. This includes all hardware, in particular the sensors, so that data can be recorded for offline analysis. No semi-automatic movement is added in this deliverable.
Software for allowing the different parts to communicate has also been added to enable synchronization of the data collected. The work done comes from Task 6.2 and Task 6.4.

Several new mounts had to be constructed and added to the Giraff platform to be able to hold the extra equipment. Two laser scanners, a gyroscope, a Kinect box for environmental analysis, a high-definition camera for facial analysis and an array microphone for speech analysis. Both the environmental analysis and the facial analysis is highly CPU intensive so two additional computers are added.
The environmental analysis is done by an embedded computer, that will also take care of the semi-autonomous movements that will be added in D6.3. The facial analysis is done on a high performance laptop strapped to the Giraff platform. Due to the fact that this laptop needs to run at full speed it cannot run on its internal battery, but it has to be connected to an external power. To this end an extra battery and an inverter were also added.

Finally tests and an evaluation of the system was done to make sure the system works as desired.

2 Contributors

The authors of this deliverable are Lasse Odens Hedman, hardware engineer at Giraff Technologies and Mårten Scherlund, software engineer at Giraff Technologies, but many others have provided invaluable help and feedback.

Lasse Odens Hedman came up with the plan for how to mount all the extra equipment that was added to the Giraff platform, constructed the mounts for the sensors and the extra power system.

Mårten Scherlund was involved in the architecture planning and set up the clock synchronize system. Kyriacos Shiarli also made a big contribution by compiling information into a schematic that helped identify many potential problems.

3 Manually controlled telepresence system

3.1 Objectives

This deliverable is going to be a manually controlled telepresence system. That means that all sensors needed will be added to the TERESA robot, but the controlling of the robot will be done manually using the existing Giraff platform. The sensors will however be used to record data, exactly as they are going to do in future semi-autonomous versions of the TERESA robot. Using this deliverable it will be possible to see if the sensors are placed correctly and that valid data can be collected from them. As the sensors are connected to three different computers on the TERESA robot an underlying architecture was developed to be able to sync the data between the computers. So the objectives for this deliverable are:

- Construct mounts to be able to mount all sensors and other components on to the Giraff platform
- Develop a power system to be able to power all the extra equipment added.
- Develop a communication architecture for the different components to be able to synchronize with each other.
- Test and evaluation of the manually controlled telepresence system.

3.2 Development of power system

Transforming the Giraff platform into a TERESA robot will include adding several sensors and computers. The battery of the Giraff platform is already fully used by the Giraff platform itself and therefore a second battery will be added. Also the laptop, that will be running the facial analysis, needs to be connected to an external power source to run at full speed. In order to deliver “wall power” while being on a robot that moves around wirelessly, an inverter needs to connected to the second battery. The inverter is a device that converts a direct current to an alternating current, in this case 230 VAC. Table.1 shows all devices that are added to the Giraff platform and its power consumption.

Device	**Count**	**Power consumption**	**Total**
Laptop	1 pcs	230 W	230 W
Embedded computer	1 pcs	50 W	50 W
Laser scanner	2 pcs	3.6 W	7.2 W
Kinect	1 pcs	14 W	14 W
Gyroscope	1 pcs	0.6 W	0.6 W
Array microphone	1 pcs	2.5 W	2.5 W
High-definition camera	1 pcs	7 W	7 W
		TOTAL:	311.3 W

Table.1: Power consumption for all devices on a TERESA robot

From Table.1 we can see that we will need a battery that can deliver 311.3 W continuously for a desired time. We are going to use three TERESA robots to alternate between at MaDoPa, where the trials are going to occur, so an estimated battery time of around 20 minutes are desired.

Because most of the devices that will be added to the TERESA robot runs on 12V we want a 12VDC battery and therefore we need a current output from that battery around 26A to reach an output of 311.3 W. The weight is also a factor to take in account to avoid problems with the suspension of the Giraff platform. From these requirements a 12V 15Ah (180Wh, 30A rate) LiFePO4 battery is selected. This battery will give the TERESA robot a battery life of about 30 minutes.

3.3 Construction of mounts

The sensors are all going to be used for different things and therefore needs to be placed accordingly. Heavy objects like the extra battery should be placed as low as possible to avoid changing the balance of the Giraff platform.

3.3.1 Laser scanner, Kinect and gyroscope mounts

The laser scanners and the Kinect are going to sweep around the robot to detect objects that it might run into, therefore they need to be placed so they can work together as well as possible. The laser scanners and the Kinect will also be used together with the gyroscope to be able to navigate semi-autonomously.

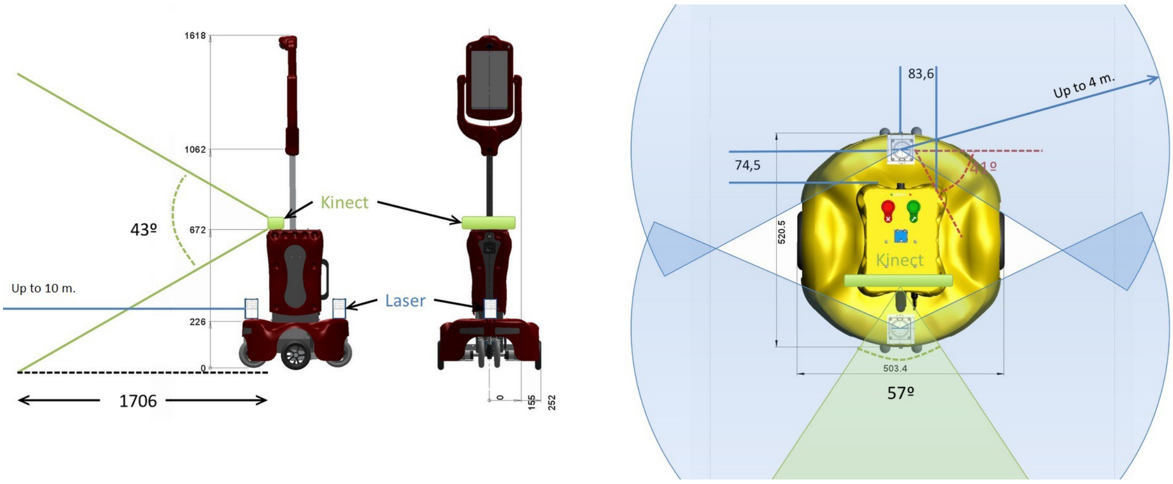

Fig.1 Placements of laser scanners and Kinect sensor

As can be seen in Fig.1 the laser scanners are placed low to be able to detect most of the objects that might appear in the different scenarios described in D6.1, such as people, walls, tables and chairs. They will also be able to track a person walking next to or even behind the robot due to the scanner pointing backwards. The laser scanners are pointing in a horizontal direction on a fixed height, so objects that are lower or higher that that fixed height won't be detected. The Kinect sensor, with its wider area of detection, will help with the detection of objects when the robot is going forward. The full coverage of these sensors are shown in Fig.1. The gyroscope, which is not visible in the figure, is placed horizontally behind the Kinect sensor.

3.3.2 High-definition camera mount

The high-definition camera is going to be used for facial expression recognition and therefore it needs to be placed at a height where it can see the face of people of different height clearly. The Giraff platform's moving head is a delicate construction and no more weight can be added to the tilt mechanism. For this reason the camera needs to be placed below the LCD screen. To be able to handle both sitting and standing persons the mount needs to be able to be tilted up and down. Fig.2 shows the mount placed in a horizontal position just above the array microphone mount.

Fig.2 High-Definition camera mount and array microphone

3.3.3 Array microphone mount

The array microphone is supposed to pick up the conversation from the room. Therefore it should be placed high up to be closer to the height of peoples mouths. Fig.2 shows the array microphone placed under the mount for the high-definition camera.

3.3.4 Embedded computer and laptop mount

The laptop, which is of workstation type, weighs about 4kg, which is heavy for the Giraff platform and therefore it is placed as low as possible on one side of the robot. It is fastened with a strap. Fig.3 The embedded computer is of the same size (mini-ITX) as the on-board computer of the Giraff platform and therefore it is easy to mount it on the outside of the box in the same mounting holes.

3.3.5 Battery and Inverter mounts

The battery is quite heavy and therefore placed as low as possible to avoid changing the balance of the Giraff platform. It is secured with a strap at the top of the base of the robot. The Inverter is placed above the embedded computer and it is also fastened with straps. See Fig.4.

Fig.3 Laptop mounted

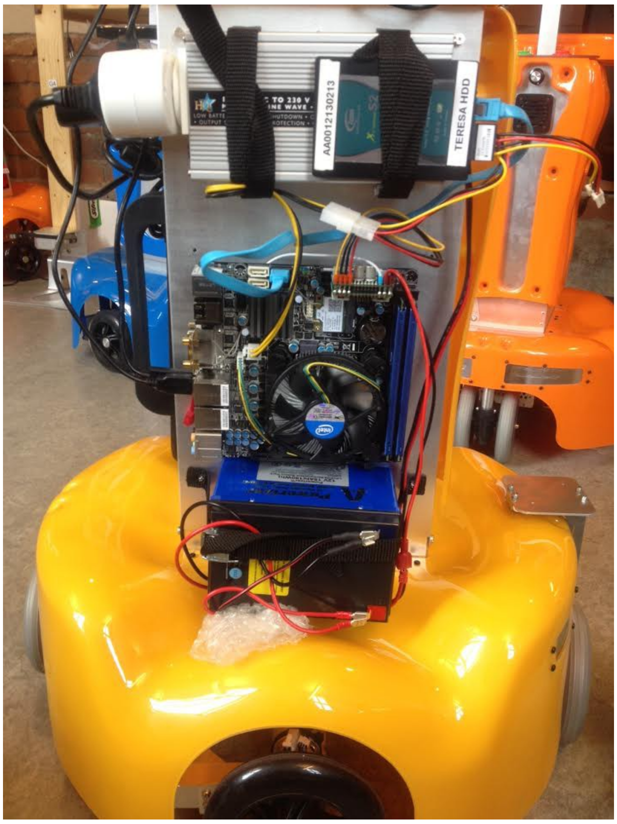

Fig.4 Battery, embedded board and Inverter

3.4 Development of software architecture

The fact that the TERESA robot consist of three different computers that will have to communicate with each other means the system will require an overall architecture. To be able to share data in a usable way the computers clocks also needs to be synchronized with each other and such system is also included in this deliverable.

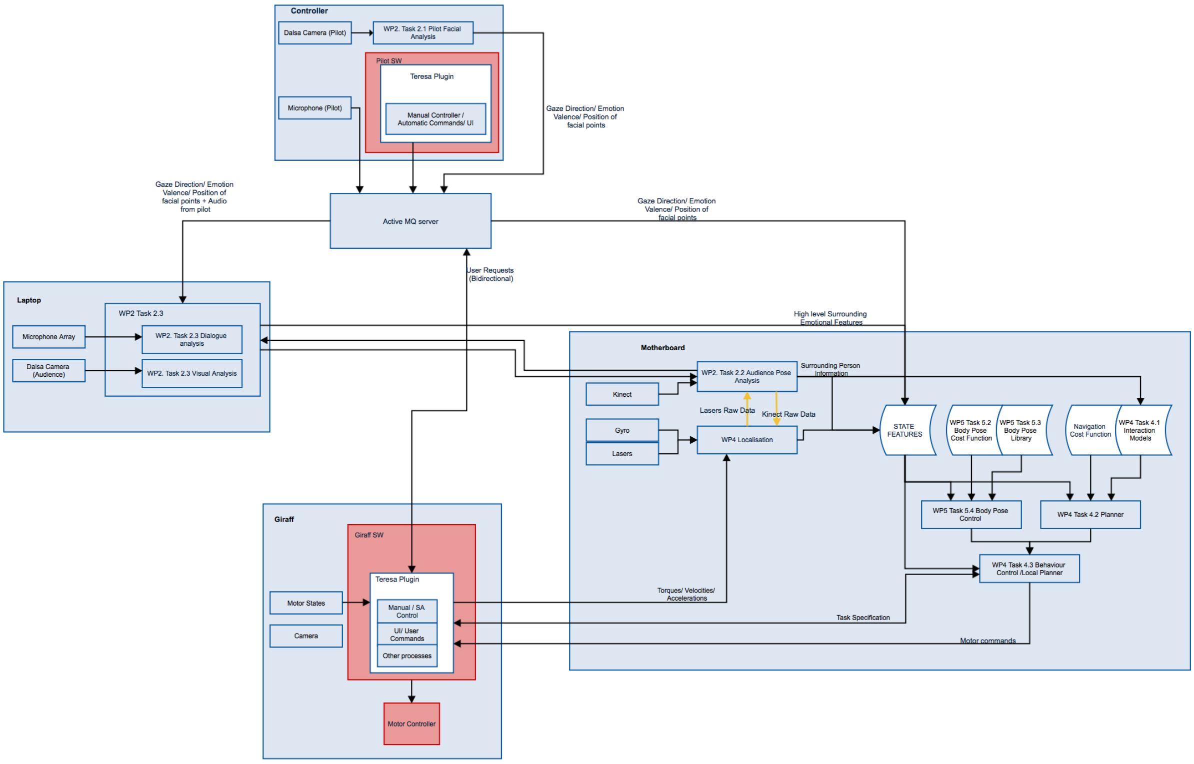

Fig.5 Communication architecture. See Annex 1 for bigger version.

3.4.1 Communication architecture

The software components of the Teresa project will primarily exchange information using ActiveMQ. In the 0.0 version there will be an ActiveMQ server running on the Giraff. It has been proposed that components use the MQTT protocol for data exchange, but this is not a requirement.

Preliminary analysis indicate that only processed information (as opposed to raw sensor data) will be exchanged and for that reason the data flows are estimated to be small compared to the video streams between Giraff and Pilot software and we forsee no network performance problems.

In version 0.0 the infrastructure for communication will be in place, but components are not required to exchange data at this stage as we will only dump raw sensor data to disk.

3.4.2 Computer clock synchronization

Version 0.0 will be used to dump data to disk and to get usable data the clocks on the three involved computers needs to be synchronized. NTP will be used for clock synchronization. The linux motherboard will run an NTP server and the Giraff and Laptop will run NetTime (http://www.timesynctool.com/), which is a free (BSD License), open source windows application for synchronizing with NTP servers.

Before picking NetTime we also evaluated w32time and the Meinberg distribution of the windows port of the linux implementation of NTP. Both proved to be unsuitable for our needs as they do not correct drift often enough. As we need to maintain a sync in the millisecond region during data collection we need to sync often. This is the killer feature provided by NetTime.

With NetTime all three computer have consistently been synced within 5 millisecond over several hours.

3.4.3 Deployment architecture

The Giraff and the strapped on extra computers will need to be updated as bugfixes and configuration changes are made. To facilitate this a Git repository has been set up (bitbucket.org) and this repository will contain software and configuration files for the TERESA computers.

As a computer boots it will:

1. Pull the latest changes to the repository.
2. Check out the appropriate tag in Git (configurable).
3. Run the startup script for that computer. This script is located in the Git repository.

Using Git will allow updates without needing to connect a monitor, mouse or keyboard to the computer in question. There will be a “production” tag that should be the latest and greatest version. In addition partners are free to create whatever branches and tags they need to do their work. By changing the bootup script TERESA computers can be instructed to use another tag/branch, which will be useful during development.

This scheme is designed to allow partners to work and do tests independently, while at the same time providing an easy way of distributing fixes and features to all sites once completed.

3.4.4 Hands on remote access

Anticipating that sometimes using Git will not solve all problems a piece of software called TeamViewer (http://www.teamviewer.com/) will be installed on all TERESA computers running Windows 7. It is free of charge and allows remote access to a Windows desktop.

The key feature of TeamViewer is that it most cases it is possible to connect remotely without changing any network configurations.

For the linux motherboard a different solution might be needed, but so far it is assumed that sshd will be running and that a commandline interface will be sufficient.

3.4.5 Documentation

Part of this deliverable was documenting how to set up the various TERESA computers. There is a written instruction for each computer and this instruction is checked into the Git repository together with all installation files required to get the TERESA system up and running.

There is a catch-22 however; Git itself must be installed prior to cloning into the repository. Instructions on how to install Cygwin/Git will be in the repository, but must also be distributed separately.

The documentation has been added to a Git repository hosted at bitbucket.org. For access send an email to marten.scherlund@giraff.org as it is currently invite only.

3.5 Test and evaluation of the manually controlled telepresence system.

3.5.1 Weight and balancing

The Giraff platform have two powered driving wheels, one on each side of the unit. Those two together with two balancing wheels, in the front and back of the unit, are what makes the Giraff platform able to keep its balance. To be able to climb thresholds the powered drive wheels are attached to a suspension system, which helps keeping the wheels pressed to the ground for good traction. This pressure is a quite delicate setting because if you put too much pressure on the drive wheels the Giraff platform gets wobbly when moving around and it can even tip over. Too little pressure on the driving wheels will cause most of the weight to end up on the two balancing wheels and this makes the Giraff platform loose traction to the ground and hard to control.

The TERESA robot has close to 10kg extra equipment added and this affects this settings. When driving around with the standard spring damper, which now is too weak, the platform's wheels slips and the TERESA robot has a hard time moving around due to the overweight. To solve this a new stronger spring damper was needed, replacing the standard one. Finding the correct strength was done by trial and error until a suitable spring damper was found.

3.5.2 Battery time.

To test the length of the battery time all components should be added and their power consumption should be at maximum. Due to some problems with delivery times a couple of sensors could not be added. These are the two laser scanners, the high-definition camera and the gyroscope. All these components are low powered and the test should give a satisfying result close to reality anyway.

To get the two extra computers to have a heavy workload and consume more power, both prime95 and 3DMark11 is loaded on them. Prime95 maximize the CPU cores and 3DMark11 includes a lot of graphics calculations so the Graphics processing unit (GPU) get to work. But the GPU is not used at its maximum so therefore the power consumption is less.

The test is started and is declared finished when the battery runs out of power and the laptops shuts off. The result is that it runs of of power after 64 minutes. As said earlier the reason for this long battery time is that no software available at the moment for the test was able to maximize the power consumption on either of the two laptops.

4 Conclusions

The first version of the TERESA robot is constructed by adding mounts for the different sensors and computers and a communication architecture for them to communicate. The sensors were placed where theoretically they would perform as good as possible, but the first test at the facility in France will show if changes are needed for the next version of the TERESA robot.

The weight and the potential balancing problem was taken care of by making sure to place the heavy objects close to the base of the Giraff platform. To preserve the driving capability a new spring damper had to replace the current one to accommodate for the higher weight. Further testing with the upcoming autonomous driving will have to show if more adjustment to the spring is required, or if some value in the control loop will have to be adjusted to help the TERESA robot to move around with its higher weight.

Architectural discussions led to the creation of an architectural overview depicting how hardware and software will be wired together. Several issues surfaced in the process and we now have a good picture of how the TERESA robot's hardware and software components will interact.

Software was added to the TERESA robot's three computer to enable clock synchronization, data exchange, remote access and unsupervised TERESA software updates. Of these only clock sync is actually required for this deliverable, but remote access and software updates will make life easier already at the first experiment.

Annex 1 – Communication architecture