

Question #	Interviewee 1	Interviewee 2	Interviewee 3	Interviewee 4	Interviewee 5	Interviewee 6	Interviewee 7	Interviewee 8	Interviewee 9	Interviewee 10	Interviewee 11	Interviewee 12	Interviewee 13
1.2. Did you ever face the problem of the lack of standards in service robotics	Yes, communication with components. The problem is that there are many standards	Yes, methods and criteria for the evaluation of robot performances	There is often general confusion about the many types (levels of abstraction) of "architecture" capable of being described within a robot system. On a more practical level specific safety standards would make it easier to meet legislative requirements, but it is probably too early to formulate these.	yes, For interconnection of HW/SW modules	no	yes, electrical architectures and power systems	No	Yes, benchmarks for performance evaluation in cognitive robotics	Yes, middleware, communication, benchmarking, everywhere	yes, safety requirements	yes, SW APIs	Interfaces and common prog. Languages	safety
2.2. Which enhancements should be done with respect to scalability and technical soundness of the mechanisms used for implementation	Open source, stable implementation, portable code	no answer	Again across the wide variety of robot applications only design time commonality can sensibly be sought with common functional architectures. Within similar application areas common operational architectures may be implemented offering build / compile time scalability. Only within product families / closely controlled application areas can implementation architectures be shared and plug & play scalability be implemented.	no answer	possibility to integrate controller in one network (cooperating systems)	no answer	not clear, no answer	Mostly control software, including AI (Adaptation, Navigation, Learning Human Robot Interaction etc.)	We have to start with interoperability (i.e. Putting together components coming from different suppliers), then the best architecture will take over	no answer	no answer	no answer	no answer
3.1. What products do you offer that require distributed computing or communication middleware?	None	no answer	Our control systems often employ distributed computing. Assuming that by middleware you mean media / software combination then we frequently use bought in solutions up to the Transport layer of the OSI model.	robuBOX	no answer	military robotic platforms and systems (JAUS)	mobile platforms, mobile manipulators	Webots: a SW for fast prototyping and simulation of mobile robots	All our platforms are interfaced with middleware and/or communication protocols	no answer	no answer	no answer	Kuka robots
3.2. What media, and protocols do you use for the communication between the software and hardware components of your product or application?	CANBUS, Serial lines, I2C, PROFIBUS; DeviceNet, SMP	CANBUS, Serial lines, wireless networks, Ethernet, I2C, PCI/ISA; IP, UDP, TCP, CANOpen	Field buses, CANBUS, Serial Lines, wireless networks, ethernet, PROFIBUS, PCI/ISA; IP, UDP	CANBUS, Serial Lines, wireless networks, I2C, ethernet; IP, UDP, TCP, CANOpen	Field buses, CANBUS, Ethernet, PROFIBUS; IP, UDP, TCP, CANOpen, EtherCAT	CANBUS, InterBus, Serial lines, Wireless networks ethernet, I2C, PCI, PCI/ISA, Firewire; IP, UDP, YCP, CANOpen, JAUS	CANBUS, Serial Lines, Wireless networks, Ethernet, USB; UDP, TCP, CANOpen	Wireless networks; IP, UDP, TCP, Bluetooth	CANBUS, Ethernet, I2C, Serial Lines, Wireless networks	Field buses, InterBus, Serial Lines, Ethernet, PROFIBUS; IP, UDP, DeviceNet, TCP	CANBUS, Serial Lines, Wireless Networks, Ethernet; IP	CANBUS, Field Buses, PROFIBUS; CANOpen, EtherCAT, ADS Autom	Field buses, InterBus, CANBUS, Serial Lines, Ethernet, PROFIBUS; IP, UDP, DeviceNet, TCP
3.3. What are the timing and bandwidth requirements and constraints for this inter-component and/or inter-process communication?	less than 100 Hz, less than 1 Mbit/s	in the range of 100-1000Hz, 1-10Mbit/s	in the range of 1-10KHz or higher, 10-100 Mbits/s and in the range of 100-1000Hz, 1-10Mbits/s	In the range of 100-1000KHz, 1-10Mbits/s	In the range of 1-10 KHz or higher, 10-100 Mbits/s	all the ranges possible	in the range of 1-10KHz or higher, 10-100 Mbits/s and in the range of 100-1000Hz, 1-10Mbits/s	In the range of 100-1000 Hz, 1-10 Mbits/s, Less than 100 Hz, less than 1Mbits/s	In the range of 100-1000 Hz, 1-10 Mbits/s	In the range of 1-10 KHz or higher, 10-100 Mbits/s or higher	no answer	in the range of 1-10KHz or higher, 10-100 Mbits/s and in the range of 100-1000Hz, 1-10Mbits/s	all the ranges available
3.4. What are other requirements for this communication?	Reliability, Availability	Reliability, Availability	Reliability, availability, safety, security, predictability	Reliability	Reliability, Availability, Extensibility, Safety, Predictability	All	Reliability, availability, Extensibility, Safety	Reliability, Availability	Availability	Reliability, Availability	Reliability, safety	Reliability, availability	All
3.5. Is your current solution satisfying or are there any problems which you would like to have solved in a better way?	not satisfying, bandwidth should be extended	no answer	The problems of service robot standardisation have little to do with simple control communication, which is currently well served. I would really like the standardisation work to address issues such as sensor reliability, multi-sensor integration, sensor fusion standards, quality of data issues within the system and associated lifing issues, etc	Current situation is satisfying	no proprietary solutions, application of standards, respectively standardization of existing solutions (e.g. XIRP-communication interface)	More development tools and open interfaces	satisfying	We need a bit faster communication	It is satisfying from a technical point of view, but it requires still effort to be compatible with all the user's systems	okay	no answer	no, less performance	Yes
3.6. How much effort do you have to invest in the maintenance and adaptation of your communication solution?	Some Effort in driver support for different HW	no answer	it is a normal part of our development process	Not so much, we are using commercial solutions	Too much, time and effort must be reduced	<100K pounds/year	no answer	Quite some	We are trying to impose a quite general standard to our customers. This should reduce our effort. I'll tell you in a few years	medium effort	no answer	much effort in safe and fast communication	no idea
3.7. Is it important or necessary for you to have a "common/agreed upon by many" solution for your communication needs?	No, but would be nice	no answer	not usually, but certain customers like to impose interoperability standards	with partners and customers, yes	yes	NO	No	No	Highly important	no	no answer	no answer/not clear	Yes
3.8. Are you using any of the following middleware approaches for your products or applications? Are you satisfied with its performance and functionality?	No answer	no answer	Custom approach, satisfied	custom approach, MS technologies	no answer	Custom, satisfied	no answer	Custom, satisfied	no answer	.NET conforming, custom approach, satisfied	no answer	no answer	.NET conforming, custom approach, satisfied with regards to the domain where it is applied
3.9. If you are not using any middleware, what are the reasons for that?	not needed, overkill for use in embedded systems	they are too heavy and complex	no answer	no answer	not to use producer specific solutions (MS e.g.)	no answer	no need	no answer	we are using other (SOAP+custom one)	Real time OS	R&D	no answer	no answer
4.2. What are the benefits/drawbacks of establishing widely agreed benchmarks for mobile service robots?	comparability of components, drawback is that a widely agreed benchmark will likely not be very significant	no answer	Benchmarks help to provide an objective assessment of similar techniques or components. However the interdependent nature of complex systems such as robots can yield unhelpful results if applied too rigidly. For example a task planner may perform relatively poorly compared with others when presented with good knowledge of the environment and problem space and yet excel in the presence of uncertainty due to, say, degraded sensors. Also the application specificity is again a problem here. A route planner built for a relatively featureless environment of a nuclear process cell may perform quite badly (slowly) in a feature rich environment such as a typical home.	not understood	no answer	ill-defined benchmarks will be detrimental, education is needed- make sure its like for like	make systems comparable to customers	Huge benefits: (1) measurement of performance across different approaches (2) measurement of performance over time, (3) foster research, (4) make research more visible to the media and large public. No drawbacks	No drawbacks as soon as they are good enough. We should pay attention that the developments remains optimized for the applications and not for the benchmarks.	no answer	Comparison between different products	common communication and easy startups	no answer
4.3. How useful are the following items to stimulate a benchmarking culture? (please mark and name others if appropriate)	Open architectures and simulators would be useful	no answer	open architectures, benchmarks should be available when relevant	simulators, open architectures, benchmarks should be available when relevant	simulators, open architectures	simulators, open architectures	no answer	simulators, benchmarks should be available when relevant	benchmarks should be available when relevant	simulators, open architectures, benchmarks should be available when relevant	Simulators, open architectures, benchmarks should be available when relevant	Simulators, open architectures	Simulators, open architectures, central benchmarking center

Questions:	
1.2	Did you ever face the problem of the lack of standards in service robotics
2.2	Which enhancements should be done with respect to scalability and technical soundness of the mechanisms used for implementation
3.1	What products do you offer that require distributed computing or communication middleware?
3.2	What media, and protocols do you use for the communication between the software and hardware components of your product or application?
3.3	What are the timing and bandwidth requirements and constraints for this inter-component and/or inter-process communication?
3.4	What are other requirements for this communication?
3.5	Is your current solution satisfying or are there any problems which you would like to have solved in a better way?
3.6	How much effort do you have to invest in the maintenance and adaptation of your communication solution?
3.7	Is it important or necessary for you to have a "common/agreed upon by many" solution for your communication needs?
3.8	Are you using any of the following middleware approaches for your products or applications? Are you satisfied with its performance and functionality?
3.9	If you are not using any middleware, what are the reasons for that?
4.2	What are the benefits/drawbacks of establishing widely agreed benchmarks for mobile service robots?
4.3	How useful are the following items to stimulate a benchmarking culture? (please mark and name others if appropriate)

PLOT THE RESULTS AS WE THOUGAS WE WANTED TO DO

Interviewee 14	Interviewee 15	Interviewee 16	Comments
No	no answer	No	Here most of the responses refer to SW aspects of systems. In particular this covers interfaces, languages, APIs for middleware and communication. A number of respondents also refer to evaluation criteria and benchmarks. Safety also seems to be one of the main issues that should be standardized. On the other hand 25% of the respondents think that there are enough standards out there.
no answer	no answer	no answer	The poor level of responses can be explained either by laziness of respondents or by the fact that the question is confusing/too general. But the few answers confirm that interoperability/plug and play features are key to scalability of the system. This also directly reflects the answers to question 1.2. where respondent require common interfaces and APIs for SW
none	no answer	no answer	Almost 50% of respondents confirm that they use some sort of communication SW/middleware. The rest of the respondents did not answer this question.
Interbus, serial lines, PROFIBUS, Ethernet, IP, UDP, TCP, DeviceNet	no answer	serial lines, ethernet, PROFIBUS	It once again confirms that a variety of communication HW and SW protocols are used in products. Thus one needs somehow to tackle this problem.
In the range of 1-10 KHz or higher, 10-100 Mbits/s or higher	no answer	no answer	As it can be inferred from the answers, robotic SW applications should be able to support different communication and bandwidth speeds.
Reliability, availability	no answer	no answer	Here all the interviewees stress that communication in robotics SW (an HW) should be at least of high reliability and availability. So, one has to take into account these requirements while designing a system. On the other hand security is not yet of a big concern.
no answer	no answer	no answer	Here answers range from saying that the approaches used are satisfying to the answers stating that one needs to standardize sensor related technologies. In general 7 of the interviewees would like to have enhancements to their current systems.
no answer	no answer	no answer	The maintenance efforts are normal part of the work process. As it can be inferred from the answers that there is quite some effort involved in maintaining communication SW.
no answer	It is necessary in order to be able to make ISO norm	no answer	The responses differ from manufacturer to manufacturer. In particular, for big industrial companies standards are documents to build on, whereas smaller companies claim that they can get around without them. This stresses that if created a standard should not be "thick" (tries to cover/provide everything possible even if it is not required in target application domain) but at the same time self-comprehensive to allows good implementations.
no answer	custom approach, satisfied	no answer	The answers to this question once again confirm that many companies use customized solutions. It is normal for a company to do that, but the main problem is that these solutions do not even rely on single specification/standard. This also leads to high costs in maintenance.
no answer	no answer	no answer	It is clear that, if the standard/specification is "thick" it will scare its user base. Standards/Specs should be general and as thin as possible. This will give lead to easier implementation of specific and customized solutions targeted to solve domain specific issues.
no answer	no answer	no answer	This question directly reflects the answers given to the question 1.2, where the interviewees state that it would be good to have some sort of comparison methodology, criteria and benchmarks that allow customer to decide on what to choose.
simulators	standardized open architectures	open architectures, benchmarks should be available when relevant	The responses obtained confirm that companies would like to try their product features in simulation before developing prototypes. This provides flexibility that HW does not.

Question #
<p><i>1. If you had been given the opportunity to restart the entire research on 1st January next year, given the experiences from the project so far, how would you then go about the architecture and middleware issues?</i></p>
<p><i>2. There have been more recent announcements such as CLARATY and Microsoft Robotics Studio, would that make a difference today?</i></p>
<p><i>3. There are available and used standards such as CORBA, why should they or why should they not be used in a certain way?</i></p>
<p><i>4. What is needed to make developers work efficiently after only short time?</i></p>
<p><i>5. What is the main motivation (in practice) for reusing software, or for spending effort on making software reusable?</i></p>

6. *What do the developers want from a standard?*

Questions	
	1
	2
	3
	4
	5
	6

Interviewee 1

One should consider two aspects: management and technological aspects. From technological perspective, we would use different middleware package. One of the recent packages like ICE or .NET. I would also require better requirement analysis. The other issue is the level of the developers, whether they can handle complexity of the SW chosen (learning curve is too steep)

Component oriented approach is the way to go with. I think I would like to have architectures that are more service oriented and would also introduce such new systems as MSRS and CLARAty. Actually, we are already using MSRS simulation

The reason that we chose CORBA was that it exists already for a long time, its code base is stable and is supported on many OSs. One of the main problems with it is that it has steep learning curve and heavy for some applications. But the latter are outweighed by the former in case of DESIRE.

Existence of the hardware platform

Reuse saves a lot of time, because it allows usage of tested/stable technologies. In the context of the DESIRE project there is no motivation to make code reusable, though component boundaries should be made clear here. The other aspect of the system that should be supported is logging/tracing of each component and provide simulation programs for them

We think that if there was sth reasonable out there we would use it, in particular if there was a lot of tool support for it and existence of stable implementations

If you would have given the opportunity to restart the enti
There have been more recent announcements such as CL
There are available and used standards such as CORBA, w
What is needed to make developers work efficiently after
What is the main motivation (in practice) for reusing softw
What do the developers want from a standard?

InterviewResults

Interviewee 2	Interviewee 3
<p>I would not rely on idea of loosely coupled components but rather with some architecture. In this architecture it should be possible to access different levels of component interfaces. A lot of time was spent on reimplementing existing approaches. One should have first look around and try to reuse what is out there</p>	<p>One of the main problems of the project at the beginning was that many were not knowlegable in CORBA and it took some time to get to a required level. If we were to start everything a new I would still would go with CORBA. We did not have real architecture, though there should have been one.</p>
<p>I do not think that announcement of such systems as MSRS or CLARAty would make any different to the project. Because these projects were build with different goals in mind</p>	<p>Announcement of MSRS or CLARAty would not make any difference. Particularly these systems are closed and you are limited by this. I would use simulation tools</p>
<p>If one considers such technologies as CORBA, .NET, ICE, it would make sense to use them or at least to look into them. But one of the main issues with these platforms that they are too thick and provide too much unwanted functionality.</p>	<p>I did not make a choice of CORBA, so can not say anything, but the only thing we are using is Naming service. The positive thing in CORBA that you can access components remotely through VPN. But there is a lot of overhead but we don't need real-time so it is fine.</p>
<p>A better communication between partners and better project management tools.</p>	<p>Good training in system internals, better SW structure and code base, clear interfaces. Better error handling in CORBA it is really bad. It provides a lot of redundant information</p>
<p>Main motivation for reusing SW is saving effort. But in this project there is no such drive, it is related to the fact that industrial partners are not interested in pushing reuse. If it is the case, then it won't make much sense for a university either.</p>	<p>We always try to make our code reusbale, in particular client side interfaces are mostly reusable.</p>

InterviewResults

<p>If there are good robotic specific standards I would probably use them, but they should be open and flexible. If programmer sees that the concept is correct they would use it. I would start standardizing sth small like data structures, some small scale tools.</p>	<p>In our company we have our own SW architecture, because we can have a full control of its internal workings. If there was a good standard, probably we would tailor it for our own needs.</p>
--	--

re research 1 January next year, given the experiences from the project so far, how would yo
ARATY and Microsoft Robotics Studio, would that make a difference today?
why should they or why should they not be used in a certain way?
only short time?
are, or for spending effort on making software reusable?

InterviewResults

Interviewee 4	Interviewee 5
<p>One of the main issue is that system SW architecture was finalized during the development phase but it would be better if there was sth available at the initial stages of the design. If I had a choice I would use different middleware and not CORBA conformant.</p>	<p>I would choose a partner that has experience in architecture, because we didn't have any architecture at the initial stages. I would constraint people to use one single system. It doesn't make sense to have many different systems.</p>
<p>First one has to evaluate what is out there and check whether it fits the requirements of the system. If they are good why not!</p>	<p>If this systems provide common drivers, algorithms and they should be open and reusable in the form of libraries in best case, I would go for it.</p>
<p>I would not use CORBA TAO for real time application and sometimes we need real-time (it is implemented in sockets), there is a lot of delay in communication, data strcutures are of fixed length, error handling is poor. It is good if one has heteroginiety and one needs language and OS independence</p>	<p>Pluses of CORBA are that it is portable and language independent. It is only good for abstracting RPC and Naming service we use. But it also has many disadvantages: there is not a good tool support, no checker, loggers, debuggers and development tools.Error management is really bad.</p>
<p>There should a single team coordinator, people should focus on there own tasks. It would good to have consistent ontology for units, physical quantities. Better introduction to internal workings of the system</p>	<p>Hardware should be available, better management</p>
<p>We always try to write reusable SW, but one should have some constraints to do that.</p>	<p>We always try to write reusable SW, but one should have some constraints to do that.</p>

InterviewResults

<p>If there was a good standard why not use, but there is not. Player/Stage is good though but one has to spend a lot of time to get it running. What I really would like to see is the comparison matrix of what is out there</p>	<p>I would go for a good standard, if it is just a tool then I would like to have it thin and be in the form of blackbox. I don't want to care about the integration aspects. We need some general concepts/guidelines for the architecture</p>
--	---

ou then go about the architecture and middleware issues?

InterviewResults

Interviewee 6	Comments
<p>Before starting doing anything I would check what is out there.</p>	<p>From the answers obtained one could conclude that it would make sense to have some general guideline on how to design system architecture. It does not have to be finalized reference architecture like OSI ISO but sth providing insight on internals of sub-systems for a particular application domain (considering its requirements). Additionally, we can see the discrepancies on the opinions concerning the use of the CORBA. One of the mostly mentioned disadvantages of it being steep learning curve.</p>
<p>In our company we MS technologies including .NET and MSRS</p>	<p>The main objective of this question was to see attitude of the researchers toward MSRS brought by Microsoft to the market. Two (1/3) of the interviewees consider that announcements of the new platforms would not make much difference to the project. On another hand most of respondents do appreciate simulation aspects provided by such systems. But the answers still leave out the main goal of such systems (it is briefly mentioned), these systems attempt to provide a set of algorithms/drivers with standard APIs and do not necessarily force particular architecture (In case of CLARAty there is an architecture but the public release includes algorithms and drivers)</p>
<p>We do not really bother with higher levels of the system. We helped to interface from low level to high level through MONO .NET implementation</p>	<p>From the responses it is clear that CORBA (or similar) standardized platforms should not be used where fast prototyping and real-time applications are considered. We also have to note that there is no a good tool support with CORBA implementation used (TAO). But the pluses of CORBA include code stability, OS and language independence. So one has to take into account related aspects of a system to make the right choice.</p>
<p>_____</p>	<p>In addition to standard matters as availability of HW and better project management, two important issues have been mentioned: consistent ontology and internal workings of the system. The first issue is very actual in the context of RoSta. Availability of consistent ontology does not only improve communication between team members but also helps during design time (to design consistent APIs and specs). The second issue can be linked to the availability and understandability of system specifications</p>
<p>There is no motivation for us to make code reusable. Maybe some parts of the library</p>	<p>One can conclude that the main motivation for reuse is saving effort but on another hand developer do not fully support the idea of reuse because it constraints them, also most of the reusable software is on the client side applications, whereas server side (which is in most cases provided by manufacturers) there is no reuse at all since there is no motivation to do it, it also requires huge amount of effort from the manufacturer. That is why it is not done.</p>

InterviewResults

<p>I would go for standard if there was a good one</p>	<p>As can be seen from the responses, all of the interviewees confirm that if there were some good standards (with sample implementations and a good tool support) they would use it for their needs. But to understand the situation of standards in particular domain one would like to have a comparison of either different implementation of the single standard or various standards for the same purpose. Also it would make sense to standardize small things first and that the standard should be publicly available.</p>
--	---