International Journal of Humanoid Robotics

RobotShare: A Google*for Robots

Xiuyi Fan and Thomas C. Henderson

School of Computing, University of Utah, 50 S. Central Campus Dr.
Salt Lake City, Utah 84112, USA
tch@cs.utah.edu,fan@cs.utah.edu

Knowledge representation is a traditional field in artificial intelligence. Researchers have developed various ways to represent and share information among intelligent agents. Agents that share resources, data, information, and knowledge perform better than agents working alone. However, previous research also reveals that sharing knowledge among a large number of entities in an open environment is a problem yet to be solved. Intelligent robots are designed and produced by different manufacturers. They have various physical attributes and employ different knowledge representations. Therefore, any non-standard or non-widely-adopted technology is unsuitable to provide a satisfactory solution to the knowledge sharing problem. In this research, we pose robot knowledge sharing as an activity to be developed in an open environment - the World Wide Web. Just as search engines like Google provide enormous power for information exchange and sharing for humans, we believe a searching mechanism designed for intelligent agents can provide a robust approach for sharing knowledge among robots. We have developed: (1) a knowledge representation for robots that allows Internet access, (2) a knowledge organization and search indexing engine, and (3) a query/reply mechanism between robots and the search engine.

Keywords: Robot Knowledge; Networked Robots; Knowledge Indexing

1. Introduction

Knowledge representation is a traditional field of study in artificial intelligence [24,25]. More recently, knowledge sharing has attracted research interest. Previous research has been focused on the formation of knowledge, representation of knowledge, categorization and partition of knowledge, etc. Various knowledge base structures, knowledge interchange languages, and knowledge sharing infrastructures have been developed. Knowledge sharing among intelligent agents, e.g., robots, brings more power to each participating agent as it can accomplish its jobs more rapidly and/or at a lower cost.

One problem of the previous studies was that most dealt with a closed environment and defined a specific set of knowledge representation structures and communication languages in a somewhat non-standard fashion. This limits the adoptability and the flexibility of those systems. Another problem is the scalability of

*Google is a registered trademark of Google, Inc.

these systems, as most of them are designed to operate among a small number of participants. These systems may work well in a small LAN, however, when they are adapted to an open WAN, various problems emerge, i.e., network latency, platform incompatibility, communication language incompatibility, etc. One major goal is to vastly increase the scale of robot knowledge sharing.

We would like to build a knowledge sharing framework that supports a large number of data formats, is able to scale to process large amounts of data and is accessible to a large number of robots. To avoid some of the problems of earlier approaches, we develop a web based approach for knowledge sharing among robots. Our thesis is that robot knowledge sharing will be more efficient and successful than robots learning and acting on their own. A web based open architecture helps to bring more robots into the system and will enhance their performance even more.

In the past, when we needed to know something, we would look it up in an encyclopedia or find a book on the subject. Nowadays, we turn to web search engines, like Google^{TM}[a] or Yahoo^{TM}[b], and are given pointers to a large amount of information, and we usually find what we're looking for relatively quickly and easily. The semantic web holds promise for the future in which communities of practice will share knowledge to meet their needs or solve problems. We propose to develop similar capabilities for physical robots, including humanoid robots, which act in the world and must know a great deal about it. This includes robot butlers, surgeons, drivers, hospital orderlies, homecare nurses, etc. Thus, when a robot encounters an unfamiliar or unknown object in its environment, or when it needs to know how to perform a particular task with or on an object (e.g., clean it), it will be able to query a RobotShare in order to get pointers to relevant information available in the world wide web, or it will interact with a robot knowledge ontology-based sharing community.

Humans achieve this sharing mainly through natural language: queries are words that are matched to document content. For robots, it is not clear how to achieve this, and the question arises as to what representations best facilitate robot knowledge sharing. Restricting for the moment our consideration to 3-D physical objects, a description may include: geometry, physical properties, functional use, context, and natural language descriptions. Other knowledge, e.g., task procedures, may require representation of desired forces, torques, wrenches, etc., described in an appropriate sharable representation (e.g., some form of configuration space). The development of and access to networked robot knowledge can provide the basis for very robust intelligence for robot systems.

The developed framework for our solution is shown in Figure 1. In this figure, each participant robot creates web accessible knowledge repositories; the RobotShare server harvests knowledge from each of the participant robots and then organizes and creates efficient indexes into the database. Participant robots query the

[a]GOOGLE is a trademark of Google Inc.
[b]Yahoo is a trademark of Yahoo Inc.

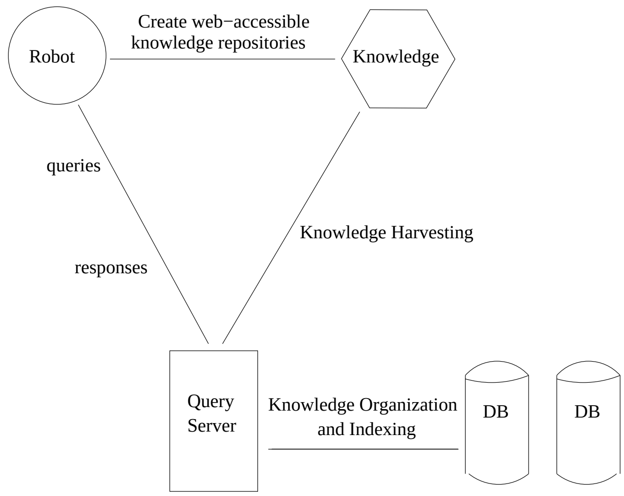

Fig. 1. The RobotShare Framework.

RobotShare server for knowledge and receive URLs pointing to other robots' knowledge. Robots, in this view, act as agents [10,21,26,27,30], and we assume their ability to generate the necessary knowledge structures; this is not an issue of investigation here.

There are at least two types of knowledge a robot may be interested to know:

- The knowledge of object identification, i.e., the knowledge to answer questions like: what is this?
- The knowledge of object manipulation, i.e., the knowledge to answer questions like: what can be done to/with this object?

Both of the two types of knowledge are closely tied to a robot's physical capacities, i.e., a robot's sensors and its actuators, as sensor data are ultimately what a robot knows about the external world and actuators define what a robot can possible do to the external world. Therefore, the proposed framework emphasizes sensor information. We believe sensor data provides a solid grounding for this research.

2. Related Work

The study of knowledge representation can be traced back to ancient Greece. Epistemology, the study of the nature of knowledge and its justification, was established by Plato in the fifth century B.C. [25]. Since then, the study of knowledge, including its nature, representation, development, etc. has been carried on by philosophers, mathematicians, linguists, and scientists. Most knowledge representation developed today is rooted in various logics. Recently, some computer scientists have expressed belief that grounding knowledge purely in logic, e.g., in symbolic languages, is insufficient for building intelligent agents and robots. They propose to develop sensor grounded and context-aware knowledge representations for robot [22,23]. Even though their work is promising, they are still far from providing a comprehensive and satisfactory solution.

Although still in its formative stages, several groups are making progress on sensor-grounded robot knowledge creation. In our provisioning effort we intend to take advantage of this. Cohen et al. [5] describe a *natural semantics* approach in which robots learn meanings through their interaction with the environment. Traditional AI approaches reply on the reduction of semantics to syntax, and such systems have no real understanding of the symbols that they manipulate. In natural semantics, such meanings are acquired and maintained by the robot system, and not specified externally by human programmers or knowledge engineers. In this work, a robot is provided with a small number of behaviors (e.g., move, turn, open gripper, etc.), and the robot records sensor data streams. From this, prototype sequences are segmented and serve as the basis for more complex tasks. In this way, the robot learns a sensor data based ontology through interaction with the environment, and concepts are related to the sense data.

Another approach is the Spatial Semantic Hierarchy which allows bootstrap learning from uninterpreted experience. This involves solving three problems: (1) *feature learning* from the sense data, (2) *control learning* for achieving desired states, and (3) *place recognition* to identify distinctive states [19]. This fits well with our robot knowledge provisioning scheme since raw data, as well as learned structures, will be available.

Grupen et al. have based their approach on human developmental theory [13]. They have demonstrated a framework for the development of robot behavior. In addition, they have proposed a relational representation for procedural task knowledge [14]. Joint probability estimates are learned which relate features of the sensorimotor stream to desired behavior quality. In this way, the robot can determine salient features in its world experiences (sensor/actuator mediated) and choose action policies. This group has examined many issues related to human-like activity (e.g., grasping, walking, etc.).

As a last example of a group producing sharable robot knowledge (there are many more, we have selected a representative sample here), Dillmann et al. have focused on robot knowledge related to their humanoid project [3]. Their recent PACO-PLUS project aims to develop a cognitive robot [8]. Their work will provide a way to bridge knowledge between humans and robots. They have recently proposed a reference model for human kinetics just for the purpose of enabling sharing. They have recently proposed a reference model for human kinetics just for the purpose of enabling sharing [2].

In order to exchange knowledge, robot agents also require a common language for the expression of their data and processes. As a starting point, common sensors and actuators give a direct mechanism for exchange. Analysis of the sensor data is then straightforward, as well as control of actuators. More abstract sharing mechanisms are possible when specific sensors and/or actuators differ; for example, Logical Sensor Systems [7,15] provide such a framework. In this case sensors are abstracted as a data type in an object-oriented sense, and physical operations by the agent on the world may be expressed as a sequence of force closures to be achieved (e.g., in terms of forces, torques, wrenches, etc.) [16].

Another influential work in knowledge sharing is the Knowledge Interchange Format, known as KIF [12]. KIF was defined as an ANSI standard by the NCITS T2 committee on Information Interchange and Interpretation in 1998. KIF is a version of typed predicate logic. It is still unclear what is the most appropriate knowledge representation format for robots, and exchanging knowledge between robots in an unrestricted environment is still a problem to be solved.

We are aware that currently there is a lot of ongoing research on the Semantic Web [6], which is led by the World Wide Web Consortium. The aim of the Semantic Web project is to create a universal medium for information exchange by putting documents with computer processable meaning on the World Wide Web. Using the Semantic Web, information can be better organized and more accurately delivered to a human reader. The book by Davies et al. [6] provides a very clear review of methods and tools developed for the human semantic web, including methods to extract information from text, retrieve information from other sources, and to compress, visualize and disseminate information.

3. Knowledge Representation

3.1. *Robot Knowledge*

Humans recognize the external world first through sensory organs. This information comes in various formats, some of them are more accurate than others; some of them are more abstract than others; some of them have temporal properties while others do not. It seems there is enough evidence for us to believe that sensory information, i.e., information collected by our sensory organs, is the ground for all of our object recognition process.

We believe robots can behave similarly, and that the best way for a robot to recognize objects is through sensor data. We would not deny ontology information like *a fork is kitchenware; kitchenware is a tool used for dining or relevant purpose; a tool used for dining or relevant purpose usually is an artifact; an artifact is an object* could be helpful at some place in our intellectual system.

We are aware that currently there is a good deal of research dedicated to various aspects of image databases [4,20] to find methods for object segmentation, object tracking, etc. Albeit interesting, these topics are beyond the scope of this research. We assume robots are able to identify objects in the environment (i.e., segment them in sensor data) and measure physical properties of the object using on-board sensors.

The definition of knowledge is still fuzzy at this point, as philosophers love to debate on this type of topics. The classic definition, found in Plato, states that:

three criteria define knowledge: knowledge needs to be a statement, such that it is justified, true, and believed [29]. For the purpose of this research, we restrict the scope of robot knowledge to be: strings, which contain information about objects and activities. This includes physical properties of objects and verbal descriptions, which are usually assigned by a human to objects; and strings that contain information about activities that includes verbal descriptions and activity components recorded in temporal sequence, i.e., trajectories. Physical properties of objects are present in various forms, some of them are temporal, e.g., acoustic information, while others are static, e.g., curvature of a surface.

We believe robots are built to help humans perform certain tasks which are either impossible or inconvenient for humans to perform. Therefore, if it is beneficial to have a robot know there are similarities between a fork and a spoon, i.e., the distance measure between a fork and a spoon is less than a fork, say, a chair; then we should program a distance evaluation function and make it to run on a robot, which always returns a smaller number when a fork and a spoon are compared than a fork and a chair. In such a measure, we define that identical objects have a distance measure equal to zero; and this measure only returns non-negative numbers. If a robot, or some programs running on the robot, could use this function to perform tasks better, then we may say that the robot knows, or is able to infer that forks, spoons and probably dinner knives, belong to one group while tables, chairs and bookshelves belong to another.

In a nutshell, *robot knowledge* is information about objects and activities, in string format, stored in robots' memory. For a robot, to know an object or an activity means to have information about that object stored in its memory. Relations between objects help robots to perform tasks better. However, relations are developed or discovered based on information acquired through sensors.

3.2. *Knowledge Extraction*

In the previous section, we have presented our definition of robot knowledge. We have emphasized that grounding knowledge to sensor data is essential to this work. This section introduces how knowledge can be extracted from sensor data, and presents data sample set that has been used for this work.

3.2.1. *Data Type and Extraction*

Sensors produce data in many formats. Typical sensors available to robots are sonar, laser range finder, weight scale, CCD camera, infrared, odometers, etc. In general, these sensors can produce results in two categories: direct measures and derived measures. For instance, a weight measure of an object is a direct measure of a weight scale; an RGB histogram of an image of an object is a derived measure of a CCD camera. The accuracy of direct measures depends on the accuracy of the instrument, i.e., the sensor; and the accuracy of derived measures depends on the accuracy of the instrument and the algorithm used to produce the measure.

Therefore, it is our hope to set standards on algorithms used to produce derived measures, or regulate the format of these measurements, so comparable results can be obtained.

To test the suitability of the direct/derived data measure taxonomy and the possibility of the measurement standardization, and to provide a solid ground for our research, we collect sample data for objects and measure their properties. Since the prototype of this framework is to be applied to kitchen robots, we collect our sample data from kitchenware, e.g., forks, spoons, knives, cups, plates, etc. CCD cameras are readily available and the produced images are information intensive. We choose to work with images as our starting point.

3.2.2. *Data Sample*

Sixteen objects, including four bowls, one cup, two forks, three knives, two plates, and four spoons, are selected. Two images are taken of each object, one from the top view, another one from the side view. All images are taken against a white background and objects are later manually segmented. Images are stored in JPEG format in the size of 640-*by*-480 pixels. For each image, the histograms of its color in RGB channels are computed, with 256 bins for each channel. The image is then converted to HSV color space, where the histogram of the hue channel is computed, with 256 bins. The Sobel edge detection algorithm is applied to compute the distribution of edge orientations, and a histogram with 256 bins is obtained. The color image is then converted to a binary image. The perimeter and the area of connected components in the binary image are computed. In order to capture the uncertainty of the real world, we have studied how these measures may vary under different picture-taking conditions, e.g., various lighting conditions. We used nature light (indoor sunlight), two incandescent lights, and a flash light, as our light sources in this test. Four objects: a cup, a plate, a spoon and a knife are imaged. Twenty images under these four different light settings are taken of each object. The RGB histograms, the hue histogram, the edge direction histogram, the perimeter and the area are computed.

3.3. *Knowledge Formulation*

In order to build the RobotShare search engine, which supports a large amount of data and fast retrievals, data indexing is needed. The indexing structure is discussed in the next section. Data format/knowledge representation is presented here.

Unlike a traditional database query system, where system architects also control the data source; or, at least, a tight connection between the data source and the database is assumed. RobotShare has the problem that it knows little about its data source, i.e., robots. It is normal for a robot to know certain features about an object; and it is willing to share this knowledge with other robots though RobotShare. In order to make RobotShare handle data from arbitrary robots, two knowledge transformations are required.

The first transformation takes place in robots, where knowledge is transformed from a robot's internal representations, which are probably only known to robots themselves, to a form such that they are understandable to other parties. The second transformation takes place in RobotShare, where knowledge is then transformed into a representation that can be efficiently indexed.

3.3.1. *The First Transformation*

The purpose of the first transformation, from a robot's internal format to an open standard, is to transform knowledge in a systematic way such that an unambiguous, widely-adoptable, format is achieved, while maintaining all information in a *knowledge piece*.

Having a clear distinction between an object instance and an object class is significant to this work for two reasons. First, RobotShare has deep roots in the concept of sensor data grounded knowledge, i.e., sensor data is the ground for all higher level knowledge structure. Therefore, knowing which instance a sensor data group refers to is important to all higher levels, e.g., semantic level, knowledge structure formation. Second, it is desirable to support instance-based query in addition to the general class-based query.

We employ the standard Extensible Markup Language (XML) to represent knowledge as the result of the first transformation, as the XML format is widely used and accessible. We give a precise definition of the language our framework supports, using the Knowledge Definition Grammar (KDG). KDG is designed to be flexible enough to capture various type of knowledge while still be parser friendly. As all knowledge about objects in our framework is sensor grounded, even though KDG provides the ability to support virtually any type of object property, we define a set of XML tags to describe certain common object properties.

3.3.2. *The Second Transformation*

The purpose of the second transformation is to convert the easy-to-communicate XML format into a representation that is easy to index. Hence we can build the search engine efficiently. We take the vector space approach.

Every piece of knowledge in our system can be divided into three parts: text data, sensor data and meta data. Text data are provided by humans. It includes the name, function, use and possible other related description about an object. Sensor data are collected through sensors. They represent physical properties of an object. They are recorded by numerical values. For instance, the weight of an object is usually recorded with a single numerical value, given a standard unit is used; the shape of an object can be recorded by a histogram of the direction of the object's edge, where a histogram is usually represented by a vector. Meta data is recorded when the object is measured by sensors. It contains information about collected sensor data. For instance, the location of where the object is encountered, the time of when the object is encountered, the type/band/model of the sensor used to collect data, etc. These three types of data can be indexed using two different approaches: different data types are either indexed separately, using multiple indexing structures; or they are combined together and we use only one uniform indexing structure to index them. We build the indexing structure in a high dimensional vector space where each object is mapped into a point in this space, i.e., every object is represented by a long vector.

The length of a data vector needs to be controlled. Dimension reduction is achieved by approximating histograms using the polynomial coefficient representation. This reduces a 256-element vector to a 5-element one. Appropriate distance measures are used to match query vectors to stored vectors. Details on approximation methods and distance function performance can be found in [9].

3.3.3. *An Example*

We conclude this section with a simple example from our data set to demonstrate the complete workflow. A robot notices an object in a kitchen. The robot measures physical properties of the object as the following: length: 12.5 centimeters, height: 5 centimeters, width: 12.5 centimeters, weight: 2.5 ounce, two images are taken on this object, one from the top view, and one from the side view. Images are segmented; color and edge histograms are computed. The robot has learned from a human that his object is a "Knife with black handle." Since the robot wants to share information about this object, in addition to organizing and uploading this data into a web space, the robot registers this information at RobotShare. It then packages the object into a RobotShare understandable XML file, and sends it to RobotShare.

After RobotShare receiving this file, RobotShare first parses the XML file and then constructs a set of vectors to capture information stored in this file. The text part, i.e., "Knife with black handle" goes through the LSI process. It becomes a four-element vector. The two images go through a sequence of image processing procedures. Color histograms and edges are produced. Then dimension reduction techniques are applied, histograms are reduced to short length vectors. Dimension and weight measures are extracted. The final result can be viewed in Table 1. Using these information, RobotShare builds an index for this data and stores it in the RobotShare database.

3.4. *Activity Knowledge*

The previous section has discussed how object knowledge can be represented. Object knowledge is one type of knowledge we would like to share, another type of knowledge is activity knowledge. Activity knowledge represents a much broader range of knowledge than object knowledge. If we consider that object knowledge is mostly about object identification and classification, then activity knowledge can be used not only for identification but also performance or execution of activities.

In order for robots to share manipulation and actuation processes for action in the physical world, we propose that specific robot mechanism reference models be developed and used as the basis for sharing. For example, classes include: autonomous vehicles, 6-DOF arms, and biologically-inspired systems. Here we use an example of a humanoid robot reference model. Note that such a reference model is analogous to the use of a universal Turing machine (e.g., Java) to share executable code on a wide variety of computing platforms. That is, each robot must run a custom interpreter to control its body to achieve the reference model specification. This research focuses on activity knowledge identification. The problem statement can be summarized as one robot records a sequence of human body movements and queries RobotShare for information to identify the activity being performed by these movements.

Collaborating with Prof. Dillmann's humanoid robot research group at the University of Karlsruhe, we have obtained data generated by the *VooDoo human motion capture system* [17,18], which gathers data of the human configuration over time, resulting in 3D trajectories for every modeled limb and joint angle of the human body. In *VooDoo*, the human body is represented by 19 4-*by*-4 transformation matrices, where each matrix describes the state of a limb joint. In each transformation matrix, the upper left 3-*by*-3 sub-matrix describes the rotation of the joint, the right most column describes the movement of the joint. (See [18] for a complete discussion of the *VooDoo* system representation.) We exploit two of these matrices: one that describes the trunk of the body transformation and the other that describes the right forearm transformation from each activity instance frame. The motion description is based on six values from each of the two transforms: 3 three diagonal elements of the rotation matrix and the three translation components. This results in 12 feature

Field	**Value**
red1:	[1.8464e-011 -6.6396e-009 4.4481e-007 3.0729e-005 -4.8032e-004]
green1:	[2.2378e-011 -8.7258e-009 8.0668e-007 8.9936e-006 -2.2540e-004]
blue1:	[2.2060e-011 -8.5507e-009 7.6566e-007 1.3780e-005 -4.5634e-004]
red2:	[1.4509e-012 2.5196e-009 -9.8969e-007 8.7476e-005 -2.4578e-004]
green2:	[3.1170e-012 1.8227e-009 -9.2036e-007 8.8420e-005 -3.8085e-004]
blue2:	[4.2376e-012 1.3480e-009 -8.7116e-007 8.9412e-005 -5.5129e-004]
edge1:	[-1.2498e-010 6.3984e-008 -1.0713e-005 6.4745e-004 -0.0063]
edge2:	[-1.3743e-010 6.9991e-008 -1.1638e-005 6.9704e-004 -0.0070]
LSI:	[-0.0509 0.2674 0.2571 0.4403]
phy_vec:	[6.0190 1.6906 11.3570 0.4998]
weight:	0.4244
filename:	'knife2a.jpg'
desc:	'Knife with black handle'

Table 1. Knowledge Extracted for a Sample Object.

vectors. We then approximate the trajectory of every feature field across frames of an activity instance by a fourth order polynomial. The end result looks similar to object knowledge that has been described in the previous sections, i.e., each activity is represented in twelve vectors that can be indexed using techniques described in the next section.

4. Knowledge Search Engine

4.1. *Robot Knowledge Search Engine*

As described by Frieden and Kuntz [11], the three main tasks of a search engine are to (1) match query keywords with related material on the web, (2) rank web documents according to relevance, and (3) provide pointers to the documents. Arasu et al. [1] set their major emphasis to be the creation of scalable index structures. Note that search engines for human created web documents try to make the linkage among web pages explicit and exploit this to create structure indexes.

4.2. *Knowledge Harvesting*

In the first generation robot search engine, i.e., RobotShare, we do not foresee a major role for web crawlers. Even if web pages exist, the meta data is not available to determine what pages to download, what is of interest in them (e.g., there are no words to count and no lexicon to help define any semantics), no popularity measure, and no standard places to find things (e.g., specific sites, in homepage, etc.). We decide to have robots register with the system and provide direct meta data and links to files.

In the previous section, we presented the format for knowledge communication between a robot and RobotShare. Since the XML file presented in the previous section solely contains the data of objects, a few extra fields are helpful for RobotShare. The most important one is the link to the web address where the original data can be found. Storing information about the robot which registered the information could be helpful as well. Therefore, three fields are added into a knowledge registration XML file, the identity of the robot, the time of this registration, and the link to the web page where original data is stored.

5. RobotShare Architecture

After reviewing the concept and related technologies of search engine, we are ready to present the RobotShare architecture. Representing objects in a vector space is a sensible approach for search engine/indexing structure construction. Here objects are effectively represented by sets of vectors, not necessarily a matrix, as these small vectors can have different lengths. A multi-index structure is used for query processing (see Figure 2). The advantages include: more elegant architecture and better performance with incomplete queries, i.e., queries with only certain feature fields containing data.

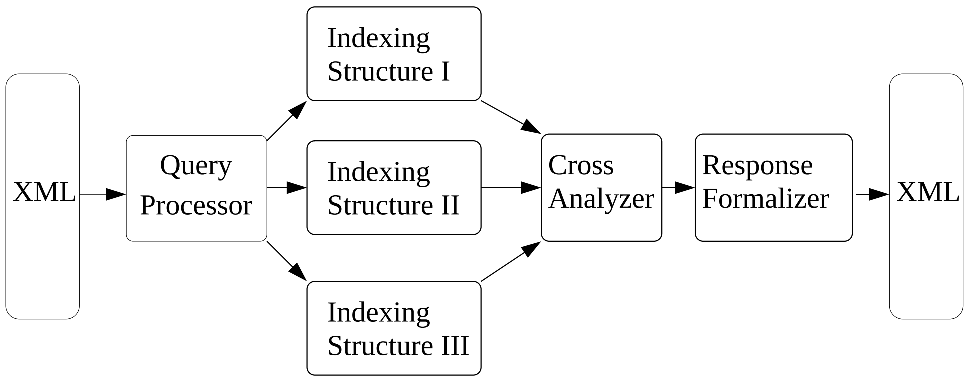

Fig. 2. The Multi-indexing Structure Architecture.

In the current implementation, RobotShare is composed of four components: a query processor, indexing structures, a cross analyzer and a response formalizer. In future, a feedback analyzer can be added to the system.

The query processor is the first component in a RobotShare query workflow. It takes queries, in the format of XML files, and translates each of them into an array of vectors and sends each vector to a corresponding indexing structure to find matches. The query processor functions as a simple XML parser as it converts data stored in XML to vectors, and computes various derived features from raw data stored in the XML file. For instance, for object knowledge, it computes color and edge histograms and transforms them into low dimensional representations. It also computes the vector representation of text information into vectors using LSI. In future, the query processor could be built with more intelligence so it not only parses data but also pre-processes them. For example, currently, robots perform the image segmentation, if the query contains an image before they query RobotShare. We may later add a image segmentation component into the query processor.

The second component is the indexing structure. It is arguably the most significant component in RobotShare. It takes inputs from the query processor in form of vectors, and produces ordered lists of objects. The objects are sorted using measures between the query sample and objects stored in RobotShare.

In the current implementation, eleven indexing components are created for object knowledge processing. Six of them are built for color histograms (two images of an object, three color channels in each image); two of them are built for edge histograms (two images for each object); one of them is built for text data produced from an LSI process; one of them is built for dimensional information of the object, i.e., length, height, width, and the cube root of the product of the three; the last indexing structure is built for the weight measure of an object. These properties are summarized in Table 2. K-d trees have been used to index all of these fields except the weight measure which uses a binary tree. In all k-d trees, branch dimension is selected in round-robin fashion, starting from the left most element of a vector. This decision is made largely due to the fact that the eight histograms in one object are all approximated by polynomials in which high order terms contribute more to the curve shape of the polynomial. Text data are processed by LSI, which has the same property that high order terms capture more information than low order ones. All indexing components return fifteen items for each query, except the LSI text indexing and dimensional properties component, in which thirty items are returned.

Component	**Vector Length**	**Returned items**
Red Channel (im1)	5	15
Green Channel (im1)	5	15
Blue Channel (im1)	5	15
Red Channel (im2)	5	15
Green Channel (im2)	5	15
Blue Channel (im2)	5	15
Edge Orientation (im1)	5	15
Edge Orientation (im2)	5	15
Text Data	4	30
Dimensional Properties	4	30
Weight	1	15

Table 2. The RobotShare Component Length Summary.

For activity knowledge, in contrast to the eleven index structure developed for object knowledge sharing, a twelve index structure is constructed for activity recognition as twelve trajectories are selected from each activity. K-d trees are used to index all trajectories. For the reason stated above, branch dimension in each k-d tree is selected in round-robin fashion. Each indexing component returns five items for each query. Items are then sent to the next module in RobotShare: the cross analyzer.

The cross analyzer takes item lists from each indexing component, “cross analyzes” them and produces a single sorted list. The cross analyzer creates the sorted list based on a weighted sum of all query-item distances. This process works as follows. The cross analyzer first creates a list contains all received items. It then computes distance from the query sample to every item in the list. The distance measure is a weighted L_1 norm, which can be expressed as the overall distance $D(A, B)$ between two objects A and B:

$$D(A,B) = \sum_{i=1}^{k} |w_i d_i(A_i, B_i)|$$

where k equals the number of fields presented in the query; w_i is the weight coefficient of the ith component; and d_i is the distance between ith components in the two objects. All $d_i(A_i, B_i)$ are computed using the L_1 norm, where

$$d_i(A_i, B_i) = \sum_{j=1}^{k} |A_{i,j} - B_{i,j}|$$

For all image histograms represented by polynomial coefficients, k equals 5; for the LSI indexed text field, k equals 4; and for the singleton weight field, k equals 1. We are at the very beginning stage to develop a systematic approach of computing weight coefficients w_i. Since the ideal ranking order can be robot and/or query dependent, it ties to the intentional use of the knowledge, the content of the database, and the content of the query template. To find the optimal weight coefficients, information about query robots must be taken into consideration. Currently, a static analysis approach is taken. We design experiments for various data conditions and query types. In each experiment we evaluate RobotShare performance using the standard information retrieval measures: precision and recall [28]. We then search for weight coefficients that maximize these measures. The search algorithm we have implemented is an n-dimensional binary search, which is a good compromise between simplicity and performance.

Once the distance between the query sample and all items returned by indexing structure are computed, the cross analyzer sorts the item list using these distances and sends the sorted list to the next module: the response formalizer.

The response formalizer takes input, which is a sorted object list from the cross analyzer, and generates an XML file that is understandable to the querying robot. The size of returned files, i.e., the length of the returned list, should be large enough so there is a high chance for the querying robot to find the information it needs in the returned file; the file also needs to be reasonably small so (a) the file transmission can be done in a small amount of time and (b) after receiving the file, a robot may determine if any useful information can be found in this file quickly. To balance these two requirements, there are two essential questions to be answered: (a) how many results should be returned? (b) What information does one result contain? In the current implementation, for object knowledge, indexing components returns either 30 or 15 items to the cross analyzer, and there are eleven indexing components in RobotShare, so the number of items generated by the cross analyzer ranges from 30 to 180.

5.1. *Experiments*

This section presents RobotShare performance on object knowledge. In all of these experiments, we have used common information retrieval performance measures: precision and recall to evaluate our system. Precision in information retrieval is defined as the following:

$$precision = \frac{|relevantdocument \bigcap retrieveddocuments|}{|retrieveddocuments|}$$

It is a measure of the percentage of results that are desired in the total retrieved list. Recall is defined as:

$$recall = \frac{|relevantdocument \bigcap retrieveddocuments|}{|relevantdocuments|}$$

It is the percentage of desired retrieval results in the entire database. Neither precession nor recall alone can indicate the performance of an information retrieval system. But by combining the two, a comprehensive evaluation can be reached.

5.2. *General Performance*

We first examine how RobotShare performs in general. Two hundred samples are randomly selected from the RobotShare database. They are used as query template to query RobotShare. The RobotShare sample database contains four hundred and eighty sample objects which are derived from sixteen real world collected sample objects, with each object duplicated twenty-nine times, with 20% random noise added to each copy. All samples in the RobotShare database, including query samples, are complete, i.e., no missing information in either queries or database samples. Weight coefficients used for distance measure, in the cross analyzer are set to one. All items returned from RobotShare are retrieved. Relevant items are siblings of the images that are duplicated from the same sample object with added noise. One result, measured in precision and recall, is presented in Table 3.

	Min	**Max**	**Median**	**Mean**	**Variance**
Precision	0.2113	0.5882	0.3704	0.3677	0.0048
Recall	1	1	1	1	0
Retrieved	51	142	81	84.8400	321.5723

Table 3. Performance Test 1.1a

We can observe that the recall for such an experiment reaches its highest possible value: one. Due to the relatively large number of retrieved documents, and the relatively small number of relevant items in the database, the precision is on the low side.

We then define the relevant items as items from the same class, i.e., bowls, knives, etc. In this setting, the range of relevant items are enlarged. This test examines how RobotShare perform on class-based queries.

From Table 4 we can see that since the average number of relevant items grows; and the number of retrieved stays unchanged, the precision grows. Due to the same reason, the recall drops.

5.3. *Activity Knowledge Experiment*

We have obtained a set of humanoid robot activity data from Prof. Dillmann's group. The *VooDoo* data representation is used here. These data contain 8 activities with each activity performed multiple times, resulting in 120 instances of activities. Since all instances are performed by a human experimenter, recorded lengths of instances range from 41 frames to 151 frames. We randomly select query templates from the activity database. Since the purpose of activity recognition is to identify human activities, Nearest-Neighbor search is more appropriate than k-Nearest-Neighbor search or α-cut search used in previous experiments. We then limit the number of returned activities for each search to be 2 (since every search always returns the query template itself as the first activity) and define the classification as correct if the second returned item is the same activity as the query template. Results are presented in Table 5. [17] indicates these results are comparable to the FFNS approach used by Prof. Dillmann's group.

6. Future Work and Conclusion

Knowledge sharing has played a major role in human civilization. The study of knowledge sharing in man-made intelligent systems, such as robots and software agents, is relatively young in the study of artificial intelligence, even though recent development in robotics, semantic web and semantic grid start to touch on this topic from various angles.

Previous sections have presented our work on the design and implementation of RobotShare, a knowledge sharing framework for robots. We have demonstrated how two important knowledge types: object knowledge and activity knowledge can be shared through such a framework. We have discussed reasons behind various method

	Min	**Max**	**Median**	**Mean**	**Variance**
Precision	0.3488	0.9178	0.6066	0.6117	0.0156
Recall	0.2500	1	0.5583	0.5668	0.0240
Retrieved	55	130	80.5000	83.2200	239.8097

Table 4. Performance Test 1.1b

Activity	**Correct**
Hold Out Hand	91.0%
Hold Out Object	95.5%
Put Object On Table	89.9%
Read Book	73.6%
Sitting	89.9%
Standing	86.3%
Take Object From Table	77.7%
Typing On Laptop	100%

Table 5. Supplemental Experiment II

selections and have compared trade-offs between different designs. The RobotShare system has been examined on a set of different data sources and we have discussed experiment results.

Future research can be divided into two categories: RobotShare refinement and RobotShare expansion. In the refinement department, a set of approaches are worth trying to discover if any of them gives better results. In the expansion department, the concept of machine-readable knowledge search engine for knowledge sharing can be taken to other domains, such as intelligent software agent. It is also necessary to rank the relevance of the returned items; this may be achieved by having RobotShare monitor the usage of the items by the querying agent (e.g., make retrieval go through RobotShare).

Another interesting topic to be studied is having the query processor to discover underlying relations between information stored in different fields in the same object. For example, when an object comes with an image and a text description as a "yellow bowl," a color histogram is computed from the image. We know there is an underlying relation between the word "yellow" and the shape of the color histogram, developing a systematic approach to discover all relations cross feature fields is an open problem. If such approach is developed, among other things, it can help RobotShare to approximate missing fields in both queries and data entries stored in its database, and possibly improve the retrieval performance.

References

1. A. Arasu, J. Cho, H. Garcia-Molina, A. Paepke, and S. Raghavan. Searching the web. Databases and the Web 37, Stanford, December 2002.
2. P. Azad, T. Asfour, and R. Dillmann. Toward an unified representation for imitation of human motion on humanoids. In *IEEE Conference on Robotics and Automation*, Rome, Italy, April 2007.
3. Regine Becher, Peter Steinhaus, Raonl Zollner, and Rudiger Dillmann. Design and implementations of an interactive object modeling system. In *IRS 2006 37th International Symposium on Robotics*, Munich, 2006.
4. V. Castelli and L.D. Bergman. *Image Databases: Search and Retrieval of Digital Images*. John Wiley & Sons, New York, NY, 2002.
5. P. Cohen and C. Beal. Natural semantics for a mobile robot. Department of Computer Science 2000-59, University of Massachusetts, 2000.
6. John Davies, Dieter Fensel, and Frank Van Harmelen. *Towards the Semantic Web Ontology-driven Knowledge Management*. John Wiley and Sons LTD, West Sussex, England, 2003.
7. Mohamed Dekhil and Thomas C. Henderson. Instrumented logical sensors systems. *International Journal of Robotics Research*, 17(4):402–417, 1998.
8. R. Dillmann and T. Asfour. Perception, action & cognition through learning of object-action complexes. Technical report, University of Karlsruhe, Germany, Project FP6-2004-IST-4-27657 2005.
9. X. Fan. Robotshare. Master's thesis, University of Utah, Salt Lake City, Utah, December 2007.
10. Jacques Ferber. *Multi-Agent Systems An Introduction to Distributed Arificial Intelligence*. Addison-Wesley, Harlow, England, 1999.

11. N.L. Fielden and L. Kunt. *Search Engine Handbook*. McFarland and Company, Inc., Jefferson, NC, 2002.
12. M.R. Genesereth. Knowledge interchange format. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 389–396, Cambridge, MA, 1991.
13. R. Grupen and M. Huber. A framework for the development of robot behavior. In *2005 AAAI Spring Symposium Series: Developmental Robotics*, Stanford, CA, March 2005.
14. S. Hart, R. Grupen, and D. Jensen. A relational representation for procedural task knowledge. In *Proceedings of the AAAI Conference*, Pittsburgh, PA, 2005.
15. T.C. Henderson and E. Shilcrat. Logical sensor systems. *Journal of Robotic Systems*, 1(2):169–193, 1984.
16. Thomas C. Henderson and Rod Grupen. Logical behaviors. *Journal of Robotic Systems*, 7(3):309–336, 1990.
17. S. Knoop, S. Brannstrom, S. Vacek, and R. Dillmann. Extraction, evaluation and selection of motion features for human activity recognition. In *Proceedings of the International Conference on Robotics and Automation*, Rome, Italy, 2007.
18. S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, 2006.
19. B. Kuipers, P. Beeson, J. Modayil, and J. Provost. Learning from experience in the ssh. In *AAAI Spring Symposium Series: Learning Grounded Representations*, Stanford, CA, March 2001.
20. M.S. Lew, editor. *Principles of Visual Information Retrieval*. Springer-Verlag, London, UK, 2001.
21. Pattie Maes. *Designing Autonomous Agents Theory and Practice from Biology to Engineering and Back*. The MIT Press, Cambridge, MA, 1990.
22. Deb Roy. Grounding words in perception and action: computational insights. *Trends in Cognitive Sciences*, 9(8):389–396, 2005.
23. Deb Roy. Semiotic schemas: a framework for grounding language in action and perception. *Artif. Intell.*, 167(1-2):170–205, 2005.
24. Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Upper Saddle River, NJ, 2003.
25. John F. Sowa. *Knowledge Representation Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.
26. V. S. Subrahmanian, Piero Bonatti, Jurgen Dix, Thomas Eiter, Sarit Kraus, Fatma Ozcan, and Robert Ross. *Heterogeneous Agent Systems*. The MIT Press, Cambridge, MA, 2000.
27. Gerhard Weiss. *Multiagent Systems A modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, 1999.
28. Wikipedia. Information retrieval — wikipedia, the free encyclopedia, 2007.
29. Wikipedia. Knowledge — wikipedia, the free encyclopedia, 2007.
30. Michael Wooldridge. *Reasoning About Rational Agents*. The MIT Press, Cambridge, MA, 2000.