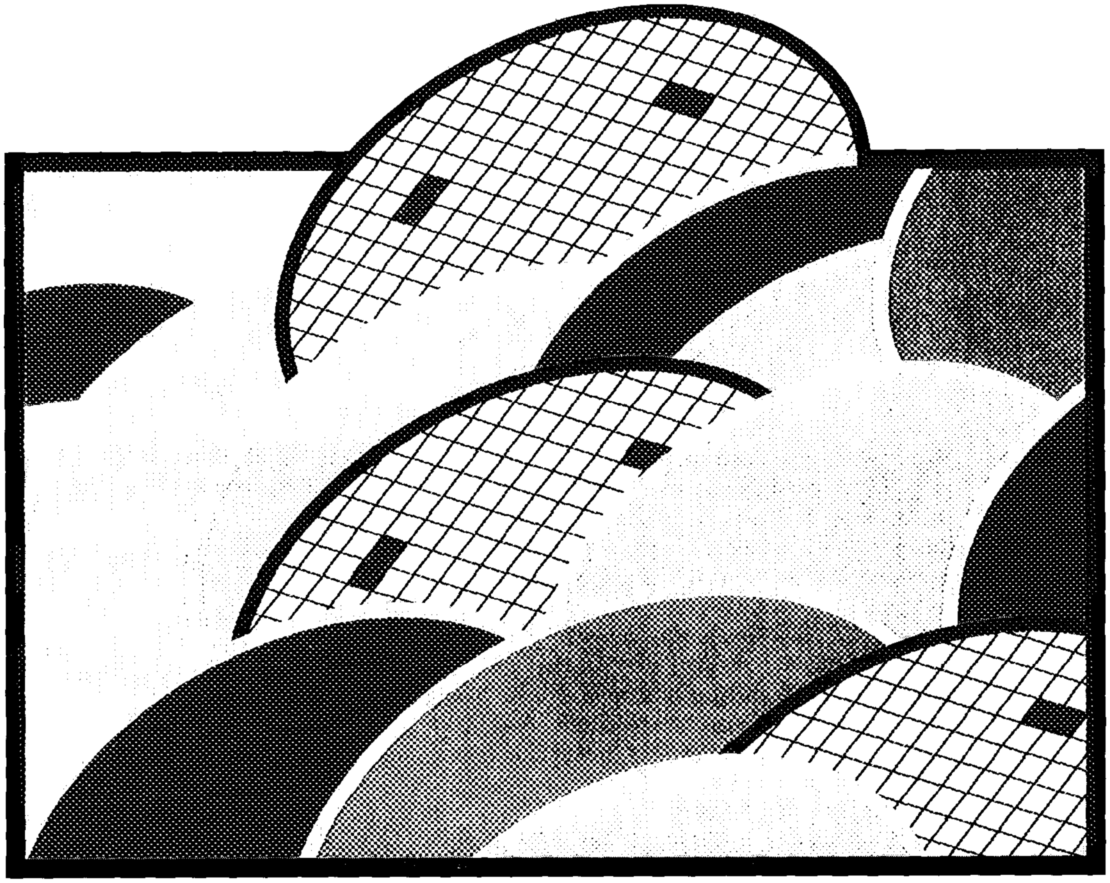
AI Magazine Volume 13 Number 2 (1992) (© AAAI)



AI Review

A Supplement to the Summer 1992 *AI Magazine*

Featuring

Delivering Lisp Applications

Nicolas Graube and Kevin Males

Harlequin Ltd.

Modern Common Lisp development environments offer unrivaled power and productivity for application development However, with a past based on specialized hardware and a present primarily focused on UNIX workstations such systems have failed to address one essential requirement: there is a market demand to deliver applications on more popular end user platforms such as PCs and Macintoshes. An effective solution to this problem must have three components:

- Compact code delivery,
- A portable user interface (UI) specification,
- Cross-platform source code compatibility

Harlequin's Common Application Programmer Interface (CAPI) has been developed to address these requirements CAPI is a portable UI tool kit that provides push-button delivery of Lisp application interfaces on UNIX/X Windows, DOS/Microsoft® Windows and Apple Macintosh. CAPI UI maps closely onto the window system of the target platform, thus ensuring natural looking interfaces (so-called "chameleon" look-and-feel). By paying particular attention to the efficiency of the implementation we have ensured that Lisp applications can compete on equal terms with conventional applications on these platforms without sacrificing the power of the Lisp development environment. Figure 1 shows an example of CAPI in action.

CAPI is a complementary solution to CLIM (Common Lisp Interface Manager). CLIM provides a rich environment with sophisticated output and input models. CAPI has been carefully designed to concentrate on tight, efficient delivery and close coverage of the underlying window system, and espouses a simpler programming model. Tools are provided to convert CAPI code to CLIM if necessary. The design of CAPI is not intrinsically language dependent and we are working on extensions to other languages.

The CAPI User Interface Model

Abstraction

In order to provide coverage of a number of different target window systems it is necessary to develop an abstract UI model which encompasses the essence of those systems. CAPI provides an abstract set of basic UI elements for interface construction. Each one of these abstract widgets is mapped onto the target window system via a library or a (CAPI) server which provides the "realization" of the abstract elements in terms of real widgets. In those cases in which an element does not have a corresponding widget, CAPI provides an efficient emulation A CAPI library must be loaded into the runtime environment. A CAPI server is a separate process that provides widget realization. Clients communicate with the server using network connections and a CAPI protocol CAPI servers can be shared between multiple clients (similar to an X server)

In designing the CAPI abstract widget set we have concentrated on the functionality and usage patterns of typical PC and Macintosh applications to ensure that the CAPI provides good coverage of those platforms.

Portability

CAPI is implemented as a common kernel together with a library for each target platform. The kernel is written in CLtL2 Common Lisp and will become ANSI compliant. It will then run without modification on any other ANSI-compliant Lisp The interaction between the kernel and library implementations is mediated by a well-defined protocol which makes it easy to add new library implementations for other window systems. The library/server interface and the abstract element set are common across all target platforms, ensuring full source code portability.

Functionality

While CAPI provides faithful coverage of target window systems, it is not limited by their restrictions. It uses Harlequin's host independent Graphics Ports system for handling all its graphical operations. Graphics Ports uses the same approach to portability as the CAPI: a small and well-defined library is required for each platform. Colors, fonts and images are all specified in a generic and platform independent way. An interface to host-specific fonts is also provided.

Delivery

Efficient delivery requires a clear distinction between the development and delivery environment In the development environment CAPI maintains a detailed description of an interface using abstract description objects that can be manipulated by meta-level tools such as interface builders, inspectors and debuggers. However none of this descriptive information is required at delivery time and so it is not transferred to the delivery environment. In general, all unnecessary code and data in the delivery environment is excised to ensure the tightest possible runtime system. The discipline of developing applications while bearing in mind the problem of delivery is one which is worth following. The design of the CAPI was driven by this need. Harlequin also supplies generic tools to address other aspects of application delivery.

The CAPI Tool Kit

Elements

The basic building blocks of the CAPI are elements. An element is the term for a CAPI abstract widget—it may or may not translate directly into a real window system element in different CAPI libraries Elements are implemented using CLOS and are essentially black boxes There are five different functional categories of elements:

- *Editor·* A group of elements which deal with text editing: A simple line editor and page editor are provided.
- *Menu:* A general menu description

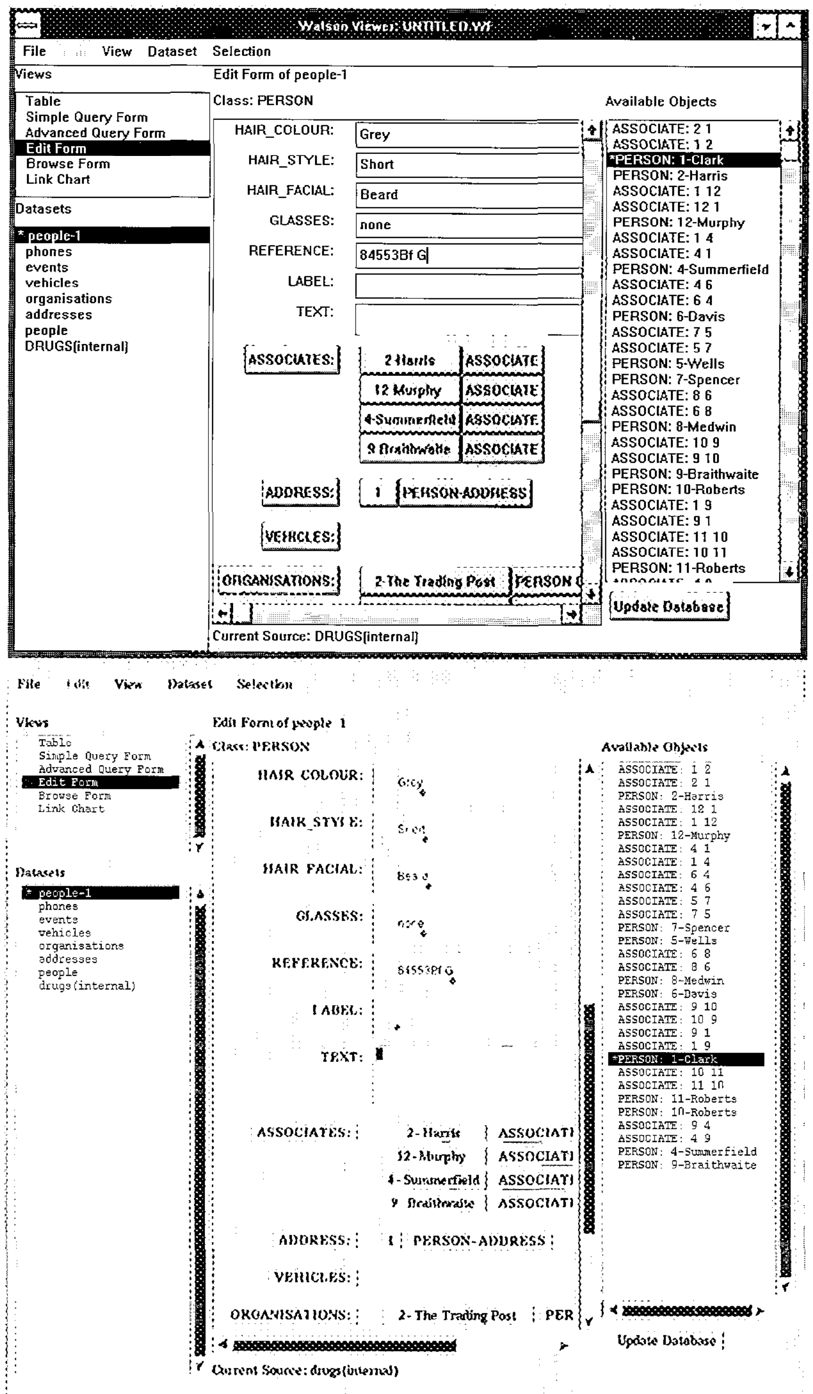


Figure 1. Capi at Work: The same application (source compatible) running under Microsoft Windows (top) and the LispWorks tool kit (bottom).

- Choice: Elements with a selection capability
- *Interaction Panes:* A group of elements on which the user can perform graphical operations and handle general input.
- *Layout.* Elements providing geometry management facilities.

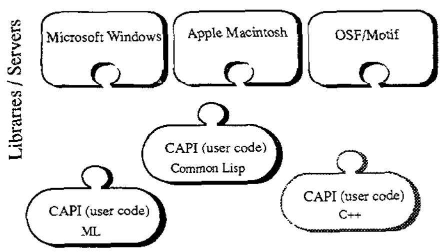


Figure 2.
Portability of the user code is guaranteed by the mediation between libraries or servers and host window system.

CAPI provides three generic operations on elements: creation, modification and display The creation and modification functions share a common interface, depending on the element type. The display operation updates an element's visual representation after (possibly multiple) modifications.

All elements that can accept user input (with the exception of interaction panes) specify their input response through one or more callback functions, according to the type of the element. Interaction panes can specify a more complex input model (see below).

Interfaces

Elements can be composed into a group called an interface using the define-interface macro. Interfaces consist of five parts:

- a group of internal states
- a group of panes. a set of capi elements or other interfaces
- a set of menus, defining pull-down menus to be attached to the top-level menu bar
- multiple layouts describing the geometrical relationships of the panes
- resources used by the application.

These groups can be specified together in the define-interface macro or separately in associated definition macros and referenced by name. This prevents unnecessary duplication of shared descriptors.

Individual components of an interface can be accessed using a functional interface, automatically generated by the define-interface macro.

An interface is also an element and hence can be used as a component in higher level interfaces. Libraries of compound user interface components can be built up and re-used in applications. For example, a color selection interface might be used in a pop-up dialog as a color prompter or embedded in a drawing tool interface.

The top level menu descriptor provides a portable means of specifying a set of menus to be associated with an interface. The visual representation of this menu depends on the underlying window system. On Microsoft® Windows and OSF/Motif™, for example, every top level interface has its own menu bar. On the Macintosh the currently selected interface displays its menu in the shared menu bar. CAPI provides a mechanism for promoting the top level menus of nested interfaces to an outermost menu of a hierarchic interface.

A powerful feature of CAPI is its dependency mechanism. The components of an interface (states, layouts, elements) can be linked together using a dependency model. Changes to one component (for example, updating a state slot) can trigger changes to other components via these dependencies.

A well designed interface should be a visualization of an application, and nothing else Both entities, interface and application, should remain separate. CAPI encourages this practice without enforcing it. In cases where the application is not easily delineated, or in very simple applications, it can be expedient to merge interface and application and make use of the dependency mechanism in both.

Geometry Management

An important design aspect of any sophisticated application interface is screen real estate management. The level of support for this facility found in underlying window systems can vary from adequate to non-existent. CAPI provides a portable layout model that defines standard geometry layouts (constraint rows and columns, for example). New layout policies can be defined by the applications programmer. Where possible these layouts are mapped into the appropriate container widgets in the underlying window system. Where this is not possible CAPI provides the functionality implicitly. Moreover within the same interface several layouts can be provided and dynamically switched back and forth depending on the application state.

Output Model

An application interface is normally a composition of standard elements and screen areas in which application-specific input/output is performed. With the Interaction Pane, the CAPI provides application-specific I/O and a portable graphics substrate called Graphics Ports with the following features:

- Full set of graphics primitives,
- Graphics state abstraction with per-port

defaults and per-call override

- Affine transformations
- Portable color model
- Generic font objects
- Generic Images

In addition to the standard screen drivers Graphics Ports provides interfaces to Postscript® and to Microsoft Windows and Macintosh printer drivers. Applications can produce printout simply by redirecting their graphics output. Output recording is available, enabling any interaction pane to replay a previous session or capture a set of graphical operations produced by an application.

Input Model

Predefined elements in CAPI have a fixed input model based on callbacks. For application-specific I/O elements (Interaction Panes) a more general input model is provided. This is based on the notion of commands which are essentially application functions. To allow for the wide variety of input devices available on different platforms (for example, a 1 button and a 3 button mouse) the mapping of user input sequences to commands is defined on a per-host basis. User input sequences can consist of mouse gestures, key presses and menu selections. The set of commands together with the mapping between these commands and host dependent gestures constitute the input model.

CAPI Applications

CAPI is already being used in Harlequin's commercial software products. Watson™ is a criminal intelligence analysis system available on both UNIX and DOS/Windows. Watson has received widespread acclaim from UK police forces and is attracting interest from intelligence agencies worldwide. The Watson interface is coded entirely in CAPI and allows us to provide both UNIX and PC versions of the product from a single code stream. Figure 3 Illustrates the Watson application on a PC and under X Windows.

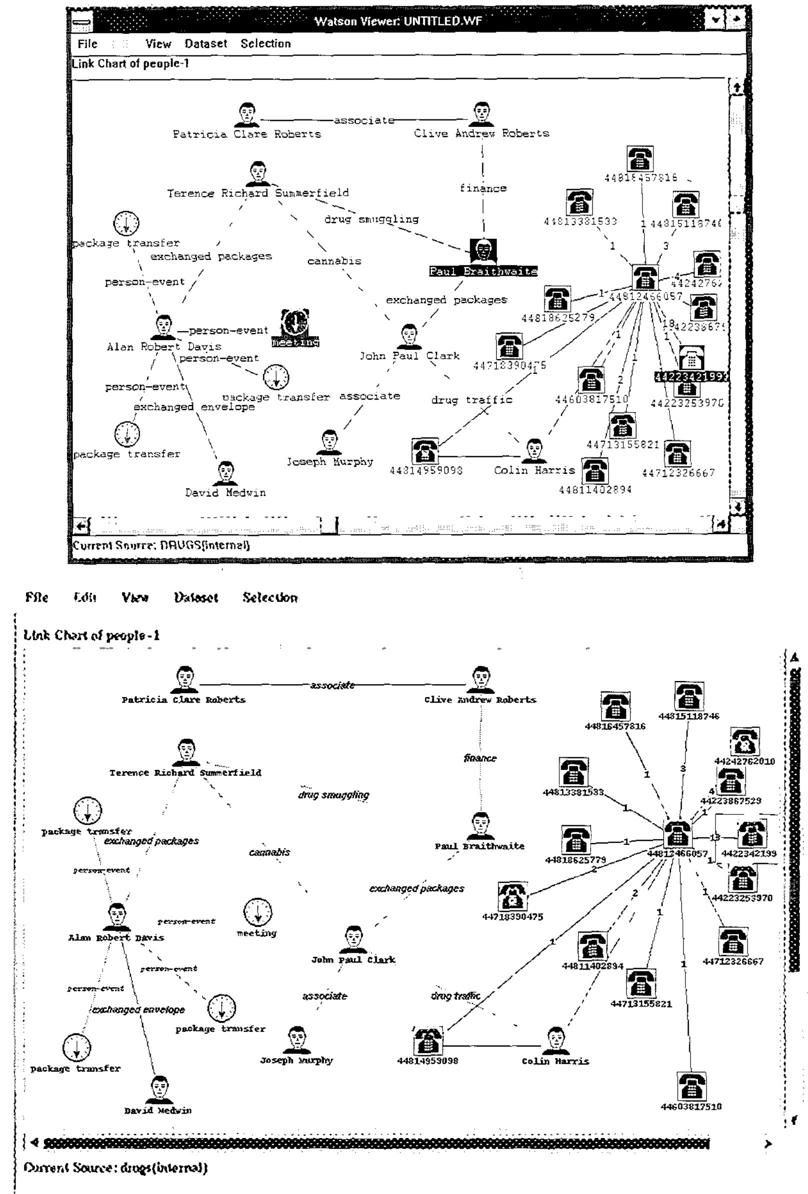


Figure 3. Watson™- A CAPI application

Conclusion

With CAPI Harlequin has bridged the gap between the power of the Lisp workstations environment and the mass appeal of PC and Macintoshes. Lisp applications no longer require their own idiosyncratic window systems to move between platforms. CAPI frees applications programmers from dealing with a number of different window systems and maintaining multiple code streams, allowing them to concentrate on using the power of Lisp and CLOS to develop the next generation of PC applications.

For more information on CAPI please contact Harlequin in either the United States or the United Kingdom:

Harlequin Inc.
68H Stiles Road
Salem NH 03079
(603) 898 1663
Fax: (603) 898 4319
works@harlqn com

Harlequin Ltd
Barrington Hall
Barrington, Cambridge CB2 5RG
+44 223 872522
Fax: +44 223 872519
works@harlqn co uk

Hybrid Systems

Tom J. Schwartz

The Schwartz Associates

For years, each new AI technology has proclaimed that it was the be all and end all of advanced computing technology This is simply not the case. In fact there is an old adage, "If the only tool you have is a hammer than all your problems look like nails "

Having attended many of the early expert system course offerings by some of the vendors this certainly seemed to be the case. In fact, one vendor, seemed to say that if the problem that I wanted to solve was not a structured selection problem (limited number of outcomes, well suited for backward chaining) than the problem I wanted to solve was positively a research issue This tended to made all the important problems "research issues "

As the AI community progressed, the idea of mixing AI technologies came forward. These have since been named hybrid systems Our firm was the first to put forth a complete hybrid systems concept in 1988 which has since evolved to Figure 1 I wish I could say that I thought of this completely on my own but that was not the case (For the purposes of this article all the components shown in the middle column of figure 1 will be collectively referred to as AI).

My first exposure to the idea of hybrid system was a bond trading system The client already had a conventional expert system doing bond trading. This system had evolved over a number of years but had only achieved a 52% accuracy. Enough to make money but not enough to be truly interesting. Our firm was asked to work on a neural net trading system for the same domain. We worked on this system and utilized the available domain experts to help us with the attribute selection and pre-processing methodologies. The resulting neural net was about 64% accurate an improvement but not an astonishing result.

However, we noticed a very strange phenomenon, the two systems were not recommending the same trades. Realizing that the two systems could be looking at different parts of the solution space someone said let's do something incredibly simplistic. Let's just poll the outputs and only make a buy when they both say yes What seemed amazing then in retrospect makes a lot of sense now, the accuracy went to about 77% and the number of trades dropped dramatically. What was going on?

What was going on was, each system was indeed looking at different parts of the solution space. By polling the outputs we were able to reduce the clients risk by limiting the trades to these where both systems agreed and there was a higher probability of success. So we can see that two viewpoints, like two heads, can be better than one.

One of the major problems in building hybrid system in 1988 was that they required a lot of hacking even when you used off-the-shelf software. Back then one had to read out an ASCII or binary file from each program after execution and then write some C code to combine outputs. Of course launching both programs and reading in the data also took, what seemed to me, too much hacking.

As more and more hybrid system came into being the problem of hacking together these different AI methods became more and more prevalent. A solution had to be found. In 1990 one was, NetLink. NetLink allowed you to hook together Nexpert Object with NeuroShell, NeuralWorks, Togai Logic's Fuzzy System shell, and AbTech's abductive inferencing machine learning tool in a seamless, reasonably easy to use manner. The only trouble with this approach was that it made Nexpert the center of the system. In fact you had to own Nexpert to even begin to use NetLink

Nexpert in and of itself was not a bad choice for the center of a hybrid system In fact, besides being a powerful expert system shell it offered one of the best set of data base interfaces available, a real strength in the construction of hybrid systems. One problem with this approach was that Nexpert is an expensive tool to buy and an expensive tool to deploy.

But wait, the year is 1991 and out of Redmond, Washington comes a real boost to hybrid system development Redmond Washington? Yes, with the release of Windows the development of hybrid systems is now within everyone's grasp How is this you may ask?

DDE (dynamic data exchange). DDE is a facility which essentially implements demons inside Windows. For instance, one can set up demons inside one program to watch values inside a spread sheet cell of another. Then when the value inside that cell changes, the changes can be linked to both a word processing program and a database program. Additionally, the demon could be used to launch other programs or processes in a multi-tasking Windows environment. Similar capability also exists in Macintosh System 7. Sounds interesting, but what does this do for you?

Think Component Software

In the past, when one wanted to link a neural net with an expert system or database it had to be done either by passing ASCII or binary files and it usually required a certain amount of hacking Alternatively, one could use NetLink to do this, however with NetLink one has to use Nexpert. Now, any product which uses DDE or System 7 can be easily integrated with any other product which uses DDE or System 7. In this way, each software package can be viewed as a component within a total solution. These components can be utilized to build complex systems as shown in Figure 1 without programming. Today there are a number of expert system, neural net and fuzzy system products which support DDE. In the future most IBM products will have to support DDE because of its ability to integrate different software packages as system components.

A Few Notes on Personal Preferences

Throughout this article, I will be referring to many software packages. In general, in product areas where there is a choice of products, I have a preference toward low cost packages

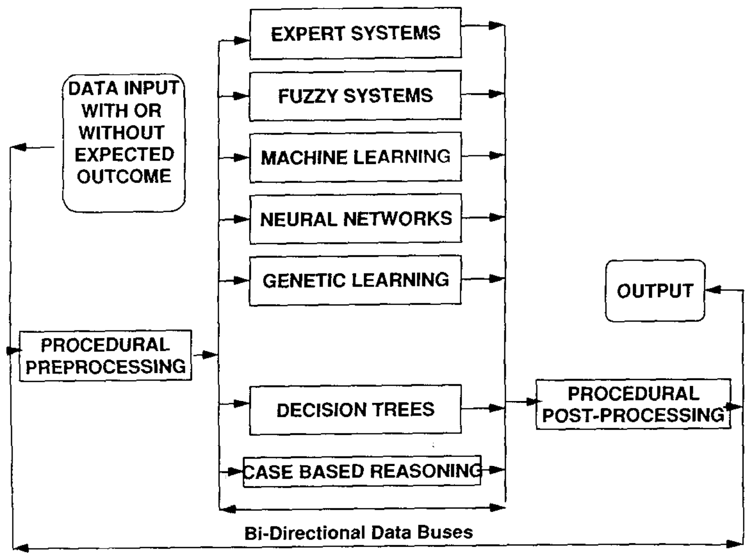


Figure 1.

from firms that offer good support. Additionally, I prefer packages which do not require a rocket scientist to operate. I belive that a very desirable goal for our industry is end-user programming. And while I don't think that programmers will or should ever disappear they should be working on more and more conceptually harder problems. This would leave the more mundane, application specific programming to end-users, who really understand the problem they are trying to solve.

It has also been my experience that flexibility brings complexity and complexity frustrates the typical user. So, if your favorite software package is not mentioned in this article it may be purely my ignorance of it (please feel free to enlighten me) or that I think that the product offering may not justify the price in most applications, Finally it might be that the product is simply too complex for the average user.

Minimum Hybrid Systems

Since it is almost impossible to conceive of a hybrid system without some procedural programming to control AI component interaction one should consider a heart or brain for the hybrid system. This should at least be a procedural programming environment which supports DDE and it may even be better if the "brain" has an expert system component. Clearly, there are many products available that fit this specification both in the programming language area and in the expert system area. There is even one that you may already own which is well suited to this task, Excel.

PROPERTY / TECHNOLOGY	EASE OF USE	REP POWER	EXPLAIN ABILITY	MIPS	RUNS ON PARALLEL HARDWARE	CHARACTERISTICS OF PROBLEM
EXPERT SYSTEMS	2-5	A-E	C-E	A-C	4-5	EXACT REASONING
FUZZY SYSTEMS	3-5	B-D	B-D	A-C	2 - 3	SOME INEXACT RULES EXISTS
NEURAL NETWORKS	1-5	A-C	A	B-E	1-2	TRAINING CASES EXIST
MACHINE LEARNING	1-4	A-B	C-D	A-E	3-4	TRAINING CASES EXIST
CASE BASED REASONING	2-4	B-D	B-D	B-D	4-5	TRAINING CASES EXIST
GENETIC ALGORITHMS	3-5	A-C	A-D	A-E	3-4	TRAINING CASES MEASURE OF FITNESS
DECISION TREES	1-3	A-C	C-E	A-B	4-5	TREE CAN BE ARTICULATED

1= EASY 5 = HARD, A = SMALL, E = LOTS

Figure 2.

The Spread Sheet Approach to Hybrid Systems

Perhaps the easiest way to construct a hybrid system is to start with Excel. Since everyone thinks they know how to work a spread-sheet and it is a very common interface this may be an excellent way to start your hybrid system. This is particularly true if your hybrid system has a neural net component since it is almost impassible to build a real neural net application without doing some data pre-processing.

Another benefit of running inside Excel and supporting DDE is the ability to fully utilize Excel macros and its programming power. In case you didn't know it, constructing a forward-chaining expert system using Excel is simple. So not only does Excel give you an easy-to-use interface and pre-and post-processing capability, it also allows one to handle symbolic variables in a forward-chaining expert system, resulting in a powerful development environment and an easy to utilize interface builder Finally, Excel offers limited file conversion capabilities so that incoming data can be appropriately formatted for the other AI tools that you may be using.

Alternatives for the Non-Rocket Scientist

Not an Excel user? One of the easiest to use expert system or should I say decision tree building tools is Adept by Symbologic. This tool not only offers ease of use through visual programming but also supports DDE. Additionally, it too provides a good environment for user interface development and basic math functions. If you are one of the rocket scientist types who think that visual programming is for wimps, Adept also offers a scripting language for hard core programmers.

Want to move up in the level of power and complexity. Level 5 from Information Builders might be a good choice. This tool offers objects, a fairly powerful expert system shell, a procedural programming language, DDE, and a very good set of database interfaces. Alternatively, KnowledgePro might be a good choice because of its good hypertext capability.

Needless to say, the list of expert system tools which support DDE goes on for a long time. The creation of a complete list is left as an exercise for the reader

Technologies

Having addressed the heart of the system let's spent a little time looking at the strengths and weakness of each of the technologies which we might use in an hybrid system.

To begin lets refer to figure 2 and discuss the property column

Clearly a chart like this is open to criticism on a wide range of fronts, however I thought that it was important to try to put all of these technologies on one chart for comparison purposes.

Ease of Use

The easy of use column refers to the currently available shrink wrap software and of course reflects the authors biases In general decision trees are the easiest types of tools to learn. This is because we have all been exposed to decision trees of one sort or another in school.

Machine learning tools such as First Class and neural net tools like NeuroShell and Braincel both use an easy to understand spread-sheet like interface and are also easy to use. One typically one has to know a little more about the problem and attribute selection to use some of these tools but they are still quite easy to use.

On the other hand expert systems typically require that one learn about forward or backward chaining. While these techniques may be reflective of what we do on a cognitive level most people are not used to thinking about problem solving in this manner.

The concept of case based reasoning (CBR) is intuitively obvious and very appealing. The tools that are currently on the market are first generation product offerings. Therefore, they will require some maturing and more automated analysis and clustering techniques before they realize the real easy of use that CBR may be capable of.

Some of the fuzzy tools are fairly easy to use, however the concept of fuzzy logic and the manipulation of fuzzy inferencing methods and defuzzification are complex topics. As a result this is not yet as easy to use technology as it may evolve to be with more product development.

The concepts behind genetic algorithms (GAs) are complex and require a good understanding of both math and the problem to successfully utilize. Additionally, only Evolver for the Macintosh is a truly industrial strength product. The tools that are available for the IBM market do not run under Windows and are designed more for hackers than they are for end users.

Representational Power

Representational power is my attempt to quantify the power and range of ways that a particular technology has to represent the information required to attack a range of problems Expert system have the ability to represent information (or knowledge) in the form of frames and rules. Many expert systems also have the ability to represent knowledge in the form or semantic nets. It is this richness of representational forms that caused me to give this technology the highest rating regarding representational power.

While there is one fuzzy system tool that does offer frame representation it is currently available only for IBM mainframes or PS/2 machines running OS 2, additionally it is very expensive and complex to use. In general, most of the fuzzy system tools offer only rule based representation and therefore limited richness.

Neural nets, GAs, machine learning and CBR all tend to represent knowledge in the form of cases or in the case of GAs constraints. While fairly powerful these techniques fall short of frame based representation. The same can be said for decision trees, however Adept with its concept of compound nodes may be moving decision trees in to the arena of more representational power.

Ability to Explain

In many applications the ability of a tool to explain its decision is critical to its deployment. For instance in the U.S. whenever a bank denies a loan it must tell the applicant why the loan was denied. This palaces certain constraints on what technology can be used in what application.

Clearly one can turn off explain in an expert system and drive its ability to explain to zero, however, this is not what the intent is in rating the explainablity of expert systems as high as I did. In fact, expert systems can, with enough programing, explain its reasoning back to a level of first principals. However, this would be very time consuming to build and would certainly not be the basis of heuristic knowledge representation.

Fuzzy systems offer much of the explain power of expert systems however following the math of the inferencing forms and defuzzification is not straight forward. This contrasts with the confidence factor methods of expert systems which are much easier to compute.

In general it is hard to get a neural net to explain itself. While HNC's KnowledgeNet can tell you which characteristics have the most positive or negative weight in causing a net to reach a particular decision it does not output rules. In fact while many neural net tools do offer sensitivity analysis this can be an awful time consuming method of deriving rules.

Many machine learning techniques output rules or decision trees which makes explanation easy. However, there are some methods like abduction which output a compound formula. While some people may feel that a formula is an exact explanation others may find a formula as indecipherable as a neural net and consider it no explanation at all. Clearly, where you stand on this issue depends on where you sit.

CBR offers explanation in the form of similar cases. While this may be intuitively satisfactory it does not offer a logical chain of cause and effect which can be followed by a person. This may or may not be satisfactory depending upon your application.

GA's can be forced to output rules in many situations. These rules certainly offer an explanation of the reasoning based on the training data. Explanation is probably not as easy in the case where you might use GA's to do scheduling optimization because of the data representation issues.

Decision trees automatically have a high ability to explain because they directly represent an articulated procedures to solve a problem.

A Caveat About Automatic Explain and/or Rule Generation

Please note that almost ALL forms of automatic rule generation (GA's, machine learning and neural nets) have no underlying understanding of the domain. As a result, they may easily make the mistake of confusing correlation for causality or producing garbage rules because of garbage data. For instance, suppose you are examining a loan history database and the system comes back and says that you shouldn't make loans to anyone from Ohio Upon examination you find that the data base had only one entry from Ohio and he defaulted upon the loan. Is the GA system in error? Probably not. Given the data it was presented with as the entire universe of the problem this was a reasonable conclusion. However because of our additional knowledge of the problem we know that this is a silly conclusion.

Potentially, one way around this would be the use of CYC (MCC's Common Sense Knowledge Base) to examine the output rules and explanations The system could than at least flag the rules that violate common sense for human review.

Necessary Compute Power

There is no question that neural nets, GA's and machine learning can be the most compute intensive during the learning phase of application development. However once the application is developed the ensuing run-time can be very fast even on fairly slow machines Additionally, the learning speed is very dependant on the data representation and the proposed solution architecture. As a result two people attempting to solve the same problem using neural nets could have dramatically different learning times based only on the number of hidden nodes and layers.

To compound this issue one should realize that some neural net learning algorithms are substantially faster than others. The Probabilistic Neural Net (PNN) is 10,000 to 400,000 times faster than standard back-propagation solving the same problem on the same machine There are also efficiencies differences amongst the various machine learning and GA methods.

In general, the necessary MIPS for both fuzzy and expert system are about the same. One time this may not be the case is in the use of hypothetical reasoning. Typically this feature is offered only in expert systems and can be a real MIPS consumer. This is because this feature causes the expert system to peruse multiple paths of reasoning based on the development of multiple hypothesis about the solution. Additionally, fuzzy systems use about 1/2 to 1/10 the number of rules of a conventional fuzzy system and tend to run faster than most backward chaining systems.

CBR can be very MIPS intensive during the case indexing and clustering operations. However this does not take the days of compute time that a neural net can consume.

Ability to Map onto Parallel Hardware

One way to address the speed problem is to map the method onto parallel hardware. This can be a real benefit for some compute intensive AI methods. For instance neural nets will map onto parallel hardware very easily, however at this time such parallel hardware is quite expensive. There are accelerator boards from some of the neural net companies, unfortunately many of these boards run only a particular firms software making them of limited value.

Forward chaining expert systems and fuzzy systems seem as if they will be more amenable to a parallel implementation than backward chaining systems. However, all of these will require some form of hand problem partitioning, a step not necessary for neural nets

Machine learning, CBR and GA's can all be made to run on parallel hardware, however once again they will almost always require some form of hand problem partitioning making them less attractive than neural nets for parallel implementation.

Decision trees are by there nature a serial tree search. As a result there would be almost no benefit to putting such a solution on a parallel machine.

Problem Characteristics

This is perhaps the most interesting set of issues we will come across in our examination of these technologies.

Expert systems require a precise articulation of the solution of the problem in rules to be useful. The development and debugging of these rules can be quite time consuming In addition, these systems tend to degrade in a very ungraceful manner when questioned about items outside the domain that the system has learned. This can be very disconcerting to the user.

Because expert system require articulated rules adding knowledge to the system after deployment requires maintenance. This inability to learn in the application environment may be an issue in some applications.

Fuzzy systems seemed to have found a real home in the process control arena. This may be explained by the fact that this domain is frequently exemplified by the knowledge that there is an imprecise relationship between some set of input variables and some desired output (like the most widgets per hour within some acceptable tolerance rate). These imprecise relationships can be naturally represented by fuzzy rules. These rules are then tuned by adjusting the class of membership function and perhaps varying the methods of inferencing or defuzzification. It is not uncommon to find that the development of the original fuzzy rule set takes only 25% of the time that is spent tuning the rule set.

One way of automating the generation of the class of membership function is to use neural nets as a classifier. This method has been used by Togai Inferlogic and other firms in the creation of class of membership functions.

Fuzzy rules also have more representational power in areas where ambiguity exists than conventional rules this results in smaller rule sets. In addition, fuzzy systems are more computationally efficient since much of what is done in a fuzzy systems is table look-up (for the class of membership function) as opposed rule inferencing which can take 10-100 machine cycles.

Automating the learning or tuning of functions in a fuzzy system after deployment faces similar problems to those faced in expert system.

Neural networks are ideal for mapping, clustering and associative memory problems. As a result, neural nets require training cases which are statistically representative of the domain of interest. These cases may also be subject to some form of pre and post-processing to achieve optimum results. In general, neural nets can learn any form of continuous mapping, however the more complex the mapping the longer it takes the system to learn.

The fact that neural nets can take so much time to learn a mapping may make them problematic for automatic after deployment learning. One might also want an expert system around any automatic learning system to make sure that what it learns is consistent with the domain.

In all case based learning methods, one should make sure that the data is accurate. It is not unusual to buy data from a publisher only to find that some of the data points are incorrect. One way to check for this is to run a clustering algorithm on the data and examine out-liers for accuracy.

One frequently stated advantage of neural nets is that they generalize better than machine learning methods. Generalization, in this context, refers to the systems facility to give reasonably correct answers to problems that it has not been trained on. This seems to be the case for neural nets, however, if the case contains data points that are significantly outside the range of the data points that the neural net has been trained on all bets are off.

Machine learning offers most of the advantages of neural nets along with the capacity to output articulated rules. One of the major disadvantages of machine learning is the inability to handle conflicting cases. Depending on the data-set that you are going to train on, this may, or may not be a problem for you.

Since machine learning outputs rules this may be a candidate technology for on-line learning in your application. This may work but once again you should have some mechanism for checking the reasonableness of rules generated by such a system.

CBR does not output articulated reasoning but rather refers you to cases which are similar to the cases that it has stored in memory. This ability to offer analogous cases can be very helpful in diagnostic and help-desk applications. Additionally, it is very easy to add cases to a CBR tool. This makes it a strong candidate for on-line learning. In this instance one might hold added cases in a reserve library which is examined each day or week by a case administrator for reasonableness. In the interim the CBR system could offer those added cases as solutions with some kind of flag to indicate that these are not yet approved cases.

GA's can also use cases however in some cases with an interesting twist. The twist, in this instance, is that the cases can be automatically generated (called chromosomes) and then tested against some form of evaluation function In the case of a planning problem the cases may be proposed routes with the evaluation function being the total miles traveled.

Alternatively, the chromosomes could be proposed rules which are then tested against a database for the creation of discrete rules. To use GA's one must be able to have an articulated measure of fitness. This could be the error rate in the classification of a large number of examples, such as one might use in a neural net or simply a mathematical function. In either case, GA's offer a lot of promise even if the current offerings in the commercial software market are both sparse and disappointing.

Decision trees require both articulated rules and the logical sequence in which they should be examined. This places a greater burden on the developer than the other technologies discussed here. This makes them idea for simple problems where a logical flow of articulated rules exist. They might also be used to control a hybrid system in place of procedural programming.

Summary

Never before has the applications developer been offered so much easy to use, easy to integrate power While this article has focused on the technologies that one might use in the construction of a hybrid system there are a few items that we have overlooked that should be mentioned in closing

1 Most of the effort in constructing an hybrid system will go into the collection of data, and interface construction. These interfaces include both the user interface and the data interface. It is almost impossible to build a useful application without getting or sending data to another source.
2. Always involve the end user in the application development and interface design Designs developed in a vacuum generally deserved to be used only in a vacuum.
3. Always check data for accuracy.
4. Temporal applications are always hard to develop.
5. Don't add technology for technologies sake.
6. Try to limit the expectations of management and the users.
7. Think BIG start small If you are going to develop a hybrid system try to partition the problem so that the first component can be used on a stand alone basis. This will demonstrate to management that you are doing something useful while getting you invaluable user experience

Tom J Schwartz founded The Schwartz Associates (TSA) in 1984 TSA provides consulting to vendors and users of advanced computing technologies encompassing: expert systems, neural computing and fuzzy systems Services include: on-site-training, videocourseware (NeuroTapes, FuzzyTapes & Expert Systems Made Easy), travailing seminars, custom and off-the shelf reports, problem and product selection, technology deployment strategies, venture capital sourcing, technology licensing and strategic marketing Mr Schwartz holds a BSEE, MSEE and MBA He is a member of HKN, TB,,, IEEE, INNS, PATCA and co-founded both the Northern California Venture Capital Association and AI Forum Inc (a non-profit corporation fro the improvement of education in advanced computer technologies) Mr Schwartz has lectured and presented for IEEE, INNS, IJCNN, AAAI, IJCAI, ASME, Stanford University, University of California, New York University, appeared on many national television broadcasts and been quoted in such publications as the *Wall Street Journal, Time Magazine Fortune* and *InfoWorld* He has written over 100 technology articles which have appeared in numerous national and international computer publications He can be reached at 415/965-4561

Give Your Team the Advantage of KNOWLEDGEWARE's™

STRATEGIC EDGE™

INTRODUCING

AN INNOVATIVE STRATEGIC EXPERT SYSTEMS COMPUTER PROGRAM DEVELOPED BY

ADVANCED SYSTEMS & DESIGN, LTD.®

Today heralds the beginning of an entirely different ball game. The way in which you compete, how you measure or evaluate plans, investments, and strategies will never be the same. After several years in development, **ADVANCED SYSTEMS & DESIGN** [illegible] proud to introduce a new series of techn[illegible] under the **KNOWLEDGEWARE™** b[illegible] **STRATEGIC EDGE™** IBM PC versio[illegible]

The key to **STRATEGIC EDGE™** is th[illegible] combines high tech, high level computer [illegible] search principles with more basic applicat[illegible] The result is user-friendly information that [illegible] relates plan evaluations, investment and ma[illegible] ing strategies in a serious, yet simplified, w[illegible]

As the task of maintaining leadership in you[illegible] business becomes more and more complex, t[illegible] advantage of **STRATEGIC EDGE™** is that [illegible] will now have state-of-the-art technology avai[illegible] to your management and staff in a sophisticate[illegible] yet friendly format.

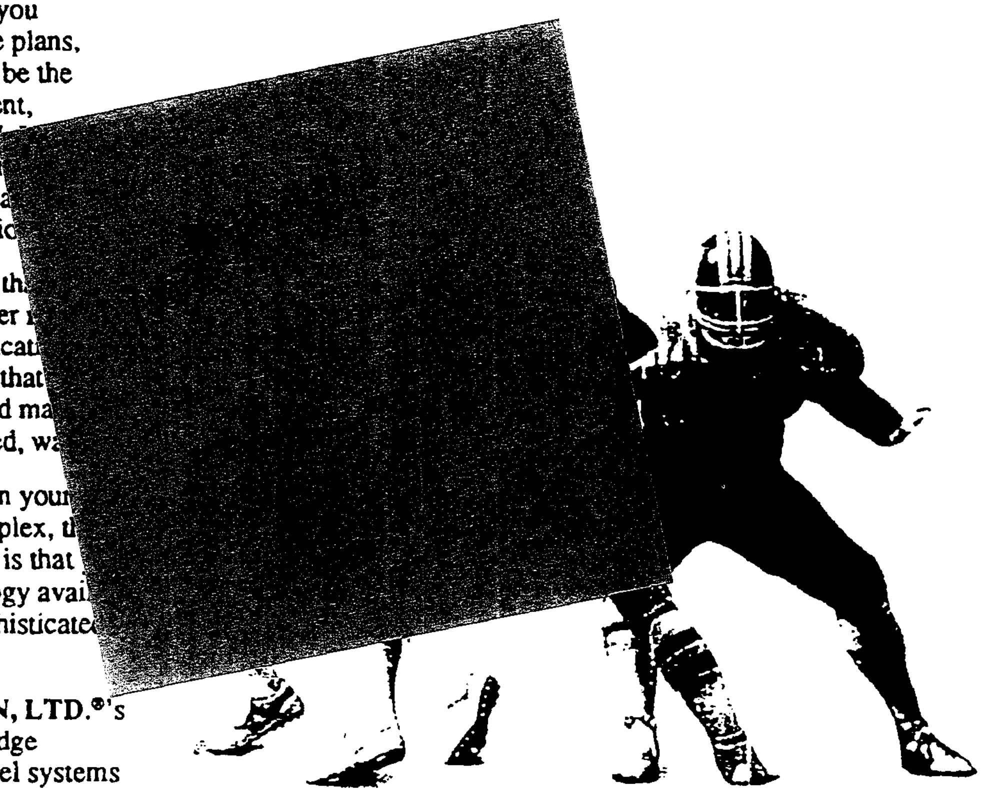
ADVANCED SYSTEMS & DESIGN, LTD.®'s Management Consultants and Knowledge Engineering Teams consist of high level systems developers who have transformed high-tech research into practical terms for Fortune 1000 clients world-wide.

STRATEGIC EDGE™ is results oriented, combined with our consulting techniques, our "Power Program" affords you an indispensible tool for planning, strategy selection, forecasting, and testing. Our recently released **OLYMPIAD™** Module further enhances this product by incorporating "Sports Strategies" into your game tactics. Our three dimensional holograph approach to planning will improve your competitive position and strengthen your *"bottom line"*.

"Stay on the offense with the winning plays in Strategic Edge™."

Give your team the competitive edge it needs with the STRATEGIC EDGE™ game plan!

Join the Revolution in Planning Technology.



Let us help you improve your Expert Systems Design, Database Development, Strategic Planning and Marketing Techniques.

For further information, please write or call:

ADVANCED SYSTEMS & DESIGN, LTD.
999 OLD EAGLE SCHOOL ROAD, P.O. BOX 38
WAYNE, PA 19087
(215) 964-8894

Automated Operations Management: The Next Generation

Dave Mandelkern

Talarian Corporation

Automating the operation of complex systems or facilities can be one of the great benefits gained from the introduction of workstations into the commercial environment. In nearly every area of business, the sheer volume of information present can overwhelm human comprehension. The profitability of a company, quality of its products, and safety of its operations are all affected by the challenge of comprehending this data in a timely manner. Even the best human experts are challenged by the flood of information they are required to analyze to make their best decisions This is especially true in real-time environments where large amounts of information must be analyzed quickly.

Real-time computer systems have become an integral part of the everyday business environment. They are being used in a growing number of applications ranging from small, simple controllers found in individual machine control environments to large, complex distributed systems for plant-wide or multi-location monitoring and control systems. The complexity of these system is increasing rapidly along three dimensions: 1. the number of functions controlled, 2. the rate at which the functions must be controlled, and 3. the number of factors that must be considered before a decision can be made.

Advances in technology and performance requirements has led to ever-increasing demands on human operators while simultaneously increasing the cost of human errors. Already some spectacularly expensive and tragic accidents, such as Three Mile Island, Bhopal and Chernobyl, have been attributed in part to "cognitive overload," or the inability of human operators to absorb and react to rapidly changing information in a timely manner.

The quest for solutions to the problems associated with managing this increasing complexity is causing considerable interest in the use of knowledge-based techniques for real-time applications. Proper application of such problem solving methods can result in more sophisticated monitoring and control strategies which support human operators in the control of complex applications such as electric power generating facilities, air traffic control centers, and process chemical plants

Real-time Expert Systems

Real-time expert systems represent a promising solution to these growing problems by combining the subtlety and flexibility of human expertise with the speed and precision of the computer. In addition to the presence of cognitive overload, factors that indicate a real-time expert system might be an appropriate solution—especially when conventional techniques have failed or are impractical—include situations where operators are unable to resolve conflicting constraints, have the potential to make high cost mistakes or miss high revenue opportunities, or are themselves an expensive or scarce resource.

Examples include an operator on an oil platform who can be confronted with 500 analog and 2500 digital signals, resulting in considerable cognitive loading in the event of a system problem. Future oil platform control rooms are being planned that will require as many as 20,000 signals to be monitored by just two or three operators. In other domains, such as satellite control, qualified personnel are becoming increasingly difficult to find, especially those who are able to evaluate complex situations and recommend actions NASA's recently launched Hubble Space Telescope has over 6,000 sensors which need to be continuously monitored in real time by human operators 24 hours a day, 365 days a year. Similarly, in the world of the stock market and foreign currency exchanges, good traders who can quickly assimilate and evaluate information and act on it are scarce and expensive.

While traditional expert system tools have been applied to certain aspects of operations automations, most existing commercial tools lack a number of key features which are mandatory for real-time problem solving, such as temporal reasoning (the ability to inference based on time histories of data), performance necessary for real-time applications, and the ability to be driven from real-time, asynchronous data inputs. Recently, a few companies, including Talarian Corporation, have developed specialized knowledge-based tools targeted specifically at the Automated Operations Management market. Talarian's RTworks product has extended many of the traditional knowledge representation methods in order to handle the real-time domain

RTworks

RTworks is a set of tools for application builders designed to address the major tasks in building an automated operator assistant system: data acquisition, data analysis (via a real-time inferencing engine), data distribution (through a client/server distributed architecture),and data display (through a sophisticated graphical user interface).

In a generic expert system shell, the inference engine, data acquisition, and user interface would typically be grouped together into one large process, potentially requiring considerable computing resources and making it difficult to react quickly to critical events. By distributing the functions of the application over independent processes, which can be executing on multiple workstations connected across a local area network, RTworks can exploit the inherent asynchrony in the system, or coarse grained parallel processing architecture, to maximize throughput and response. Such a distributed architecture also has the advantage of being able to take advantage of multiple CPUs if performance requirements call for it and also being inherently fault tolerant.

Besides the distributed architecture, RTworks has been enhanced in several other ways which allow it to apply knowledge-based problem solving methods to real-time problems. In traditional expert system shells, rules can be tested/invoked in two different ways; when data in their antecedent (IF) clause changes (this is usually called forward-chaining or data-driven inferencing), or when one of their consequent (THEN) clauses are tested to achieve a goal (this is usually called backward-chaining or goal-driven inferencing). In a real-time system such as RTworks, rules will typically be fired when a value coming from the data acquisition system changes. For example, a battery voltage drops, triggering a rule which provides a visual and audible alarm to the operator. In addition, the triggering of a rule or function can be tied to a specific time in order to ensure reliable response (check the battery voltage every ten seconds to make sure it is still within acceptable limits). In addition to ensuring reliable response times, these time-triggered rules can increase performance as the rule need not be tested every time data in its antecedent clause changes.

RTworks has extended the traditional object-oriented class representation used in expert system shells into the temporal dimension by allowing an attribute of an instance of an object to be a ring buffer where a series of values and their associated time tags are kept With a history of data now available, rules can reason about past, present, and future events. An example of temporal reasoning would be a rule which calculates the rate of change of a battery's voltage over the last 30 seconds, then calculates the time in the future when the battery voltage will drop below a critical threshold point (which in turn might be calculated from a statistical analysis of the performance of all batteries over time). Expressed very simply in an RTworks rule, such temporal reasoning would be difficult to express in a traditional expert system framework.

When a significant event occurs, it is important that a real-time expert system be able to focus its resources on important goals This concept is known as "Focus of Attention." RTworks allows several methods for focusing attention on significant events including changing the set of data the system is currently looking at, dynamically changing the set of rules the inference engine is using to analyze that data, and automatically changing the visual interface that presents information to the human operator This capability to focus attention allows the system to maintain a very large body of knowledge while applying only what is needed at any specific point in time.

RTworks has been optimized specifically for real-time monitoring and control. In many real-time applications, data is changing rapidly and rules tied to the data need to be tested and fired immediately In applications such as space operations, nuclear power plant monitoring, and process control, it is not unusual for tens of thousands of variables to be changing each second. In a typical inference cycle, data is read in from an external data source, asserted into working memory, and the associated rules are then tested and fired, possibly causing other rules to be tested. RTworks has been benchmarked at rates over 30,000 rules per second on workstations with RISC processors.

Real-World Example

An example of the application of RTworks' technology to a real-world problem is found at Pacific Gas and Electric Company, one of the largest utilities in the United States. Based in San Francisco, PG&E uses RTworks as part of its Intelligent Operational Planning Aids for Distribution Systems (IOPADS) which enables planners and operators to evaluate in real-time the conditions, such as consumption and load factor, on their electricity distribution system Taking data from physical inputs as well as a distributed database, IOPADS is designed for placement in individual substations as a distribution planning tool running on an engineering workstation It is also possible to use IOPADS to monitor and control data parameters from remote distribution substations and feeder grids from PG&E headquarters in a skyscraper office in downtown San Francisco.

Conclusion

In real-time problem solving and automated operations management, many human limitations (such as the tendency to overlook relevant information, to respond inconsistently, to respond too slowly, and to panic when the rate of information flow is too great) are most apparent, and the need to overcome these shortcomings is at its greatest The application of knowledge-based methods to real-time operations automation can result in a number of significant benefits, including reduced manning levels, training costs and skill requirements, increased safety, higher quality and throughput, less down time, and more consistent monitoring of complex systems.

Real-time systems provide one of the ultimate challenges for expert system technology. A knowledge-based system operating in a real-time situation must respond to a changing task environment involving an asynchronous flow of events and dynamically changing requirements with severe limitations on time, hardware, and other resources. During the 1990s we will see the combination of data distribution, graphical user interface, and expert system technology play a crucial role in the monitoring and control of a growing number of complex systems.

Dave Mandelkern is the President and Chief Executive Officer of Talarian Corporation, Mountain View, California Previously he was Vice President of V I Corporation, a developer of graphical user interface development software Mr Mandelkern holds a Bachelor of Science and a Master of Science (both in Electrical Engineering) from Stanford University

Exhibitors

AAAI–92 / IAAI–92 Conferences

Booth #241
Ablex Publishing Corporation
355 Chestnut Street
Norwood, NJ 07648
(201) 767-8450

Ablex will be displaying a full range of texts, monographs, and edited volumes in Artificial Intelligence. Stop by our booth for gratis copies of our two newest journals, *Journal of Computer and Software Engineering* and *Methods of Logic in Computer Science*.

Booth #441
Academic Press, Inc.
1250 Sixth Avenue
San Diego, CA 92101
(619) 231-0926

Please stop by the Academic Press booth to see our latest titles in artificial intelligence.

Booth #540
Addison-Wesley Publishing Company
Division of Higher Education, Computer Science
One Jacob Way/Route 128
Reading, MA 01867
(617) 944-3700

Addison-Wesley maintains a leading role in the publication of AI and AI-related books for both students and professionals. Please visit booth #540 to review the just published *Artificial Intelligence*, Third Edition, by Patrick Winston. This thorough revision of Winston's landmark work is available now, and instructors should be sure to reserve an examination copy. Also, visit our booth to see the following titles· *A Practical Guide to Knowledge Acquisition* by Carli Scott, Jan Clayton, and Elizabeth Gibson; *Neural Networks: Algorithms, Applications, and Programming Techniques* by Jim Freeman and David Skapura; *Intelligent User Interfaces*, edited by Joseph Sullivan and Sherman Taylor; and *Artificial Reality II* by Myron Krueger.

Booth #308
AICorp, Inc.
138 Technology Drive
Waltham, MA 02254
(617) 891-6500

AICorp, Inc., is a leading developer and provider of knowledge base software products for practical business solutions. AICorp will be demonstrating KBMS, its premier application development tool and INTELLECT, the first natural language system for data access. These products enable programmers and end-users to build and deliver "smart" applications on IBM mainframes, Digital VAX VMS, UNIX workstations and IBM PCs. AICorp will also be demonstrating a new version of KBMS featuring a comprehensive object-oriented programming capability. They are used by many Fortune 500 companies to automate decision-making processes, enabling users to make consistent, appropriate and effective judgments which incorporate an organization's policies, experience and knowledge.

Booth #539
AI Expert **Magazine**
600 Harrison Street
San Francisco, CA 94107
(415) 905-2200

AI Expert focuses on how artificial intelligence technology is used in commercial applications. Written for AI professionals—software developers, technical managers, programmers, engineers and consultants—articles cover the latest technology and practical applications. Visit our booth for complimentary issues and subscription discounts.

Booth #418

AION Corporation

101 University Avenue
Palo Alto, CA 94301
(415) 328-9595

The Aion Development System has been designed to specifically address business process automation challenges. These are applications which automate those many complex and repetitive decisions made on a daily basis which require faster and higher quality results. Technically, it is an integration of the latest developments in knowledge-based systems, object oriented application development and networking technologies. The Aion Development System builds, delivers and maintains those applications which automate business processes such as scheduling, configuration, monitoring and diagnostic, decision support and service. The Aion Development System is interoperable with many existing applications, databases, and languages and runs on leading mainframes, workstations and personal computers.

Booth #123

Alcatel Austria—ELIN Research Center

Ruthnergasse 1-7
A-1210 Vienna
Austria
+431-39-1621

The Research Center is part of the international research organization of Alcatel. Our activities focus on several topics of computer science, such as software engineering, fault tolerant computing, and knowledge based systems. In the area of expert system techniques, the PAMELA technology has been developed as the solution to the stringent requirements of industrial applications, such as telecommunication and process control systems. At the exhibition the PAMELA Development Environment is shown together with PAMELA applications: ALAMOS, an expert system for diagnosing faults in an Alcatel telephone exchange, and REPLAN, an expert system for producing schedules for a steel-making plant.

Booth #524

Andersen Consulting

69 West Washington Street
Chicago, IL 60602
(312) 507-4930

Andersen Consulting is one of the largest providers of knowledge-based systems (KBS) consulting services worldwide. Demonstrating KBS solutions to business problems developed with several key clients, Andersen Consulting shows how organizations are currently realizing the potential of KBS technology The innovative applications exhibited cover Financial Services, Insurance, Manufacturing, Government, Telecommunications, and Transportation and focus on deployed systems with very tangible results. In addition, the Institute for Learning Sciences, Northwestern University's cooperative venture sponsored by Andersen Consulting and other companies, directed by Roger Schank, will be demonstrating results of its research efforts on the use of specific leading edge technologies and methods as applied to learning and education.

Booth #228

Artificial Intelligence Technologies, Inc.

50 Executive Boulevard
Elmsford, NY 10523
(914) 347-6860

Several of AIT's Total Solutions products will be on display including Mercury ISPA—the Intelligent Statistician, MIDS—the Mercury Intelligent Decision Support System, and the Mercury Help Desk. The Mercury Knowledge Base Environment including its recently introduced Motif based GUI and improved developer's environment will also be displayed. Senior engineers will be on hand to discuss how AIT can help your organization.

Booth #240

Benjamin/Cummings Publishing Company, Inc.

390 Bridge Parkway
Redwood City, CA 94065
(415) 594-4400

Benjamin/Cummings publishes high quality books in Computer Science, Computer Information Systems, and Computer Engineering. Our new products include an upcoming revision of *Artificial Intelligence* by Luger and Stubblefield, *Concurrent Programming: Principles and Practice* by Greg Andrews, and *Conceptual Database Design* by Batini / Ceri / Navathe We also publish many titles in C++, Turbo Pascal, and other modern programming languages. We will be happy to help you at our booth.

Booth #327

Blackboard Technology Group, Inc.

401 Main Street
Amherst, MA 01002
(413) 256-8990

Blackboard Technology Group will be demonstrating Version 2.1 of GBB, its toolkit for rapidly developing and delivering high-performance blackboard applications Several prototype applications written using GBB will be running on various platforms. GBB provides a high-performance, object-oriented, blackboard database facility; knowledge source languages; control shells; and a graphical blackboard interface. GBB is an extension of Common Lisp and CLOS, and its open architecture allows you to easily integrate knowledge sources written in conventional languages and the code generated by expert system and neural network shells. GBB is the perfect tool for building intelligence throughout your entire application. Blackboard Technology Group offers a broad range of technical and educational services for blackboard-based applications.

Booth #341

Cambridge University Press

40 West 20th Street
New York, NY 10011
(212) 924-3900

Cambridge University Press will have on display at the AAAI many excellent publications. Please stop by our booth (#341) to obtain your copy of the new book, *An Introduction to the Modelling of Neural Networks*, by Pierre Peretto and many others.

Booth #238

Charles River Analytics Inc.

55 Wheeler Street
Cambridge, MA 02138
(617) 491-3474

NueX is a hybrid environment that combines the complementary power of neural networks and expert systems. The two methods complement each other such that neural networks provide soft constraints while expert systems support hard constraints. Neural networks are useful when the domain knowledge can't be easily captured in terms of rules and when there is an abundance of empirical data. On the other hand, expert systems are useful when the domain expertise can be described as a set of rules. Most real life problems fall somewhere in between these realms. NueX provides the hybrid environment that you need to handle such real world problems.

Booth #624

Chestnut Software Inc.

636 Beacon Street
Boston, MA 02215
(617) 262-0914

Chestnut Software will be demonstrating Release 3.1 of Lisp-to-C Translator. The Translator is the ideal tool for deploying Lisp based systems into commercial computing environments. The product will translate any Common Lisp application, regardless of size or complexity, into C. The resulting application is clean, extremely compact, efficient, portable C code which will run in virtually all environments, and can be easily maintained by both Lisp and C programmers. The Translator has been used to translate applications which currently operate on 15 different computer platforms. Release 3.1 (shipping 5/15/92) features among its many enhancements, a new modularized Run-Time Library. This modularization allows Lisp developers to deliver the smallest possible applications. Please stop by booth #624 for a demonstration!

Booth #633

Cognitive Systems, Inc.

220-230 Commercial Street
Boston, MA 02109
(617) 742-7227

Cognitive Systems will be demonstrating ReMind, a robust, application development tool for building a wide variety of case-based reasoning (CBR) systems. ReMind version 1.0 was released in April of 1992 on three platforms: Macintosh, OS/2, and Windows 3.0. We also plan to demonstrate a beta version of ReMind for the Sun Sparc platform at AAAI92. In addition to the ReMind development shell, Cognitive Systems also offers a ReMind C function library (API) for embedding case libraries into other applications, and a stand alone Run-time ReMind. ReMind is being used in a variety of application areas including help desks, financial analysis, diagnosis, classification, assessment and appraisal, and scheduling.

Booth #632

Covia Technologies

5350 South Valentia Way
Englewood, CO 80111
(303) 397-5740

Covia Technologies is a division of Covia, marketer and operator of APOLLO, a leading travel information management resource and reservations system. Covia Technologies' products and services include consulting, contract programming, and application products for cross industry technology solutions. CT Knowledge Engineers developed the first production Airline Gate Scheduling and Automated Aircraft Maintenance Scheduling systems. Several products can be viewed at the conference, including an automated operations system, manufacturing scheduler, airline gate and resource scheduler, an automated computer dump analyzer, and systems for the "intelligent" handling of printer traffic and data record anomalies. Demonstrations and discussions are available during exhibit hours.

Booth #234
D.C. Chamber of Commerce
1411 K Street NW
Washington, DC 20005
(202) 347-7201

The City of Washington, D.C., is a lot more than monuments and the center of the federal government: It's also a gateway to global markets, home to more than a half dozen major universities, hundreds of labs, and a fast-evolving breeding ground for artificial intelligence research and development. Washington is not only a keystone funding center, but a market center for AI applications all across the spectrum. Major AI facilities maintained by the military and civilian agencies are alive and well here as are the special research and advanced educational programs of four of our top colleges. Indeed, while D.C. is already an important incubator for AI research, it offers an equally fertile breeding ground for entrepreneurs with the capacity to develop and capitalize upon the exploding applications potential of this technology. The D.C. Chamber of Commerce's booth showcases the City's many technological assets as a location for future AI-related industry.

Booth #125
Defense Systems Management College
DSD, Building 205
Fort Belvoir, VA 22060-5426
(703) 805-5783

The Defense Systems Management College (DSMC) is a Department of Defense (DoD) institution located at Fort Belvoir, Virginia and is dedicated to providing education to the defense acquisition community. PMSS is a program management tool and a problem solving tool used by defense acquisition program managers. It is an integrated software system that is virtually hardware independent and provides a capability that integrates program management functional areas, generates program alternatives and impacts, and assesses these impacts on the functional areas. PMSS is Government owned and can be readily modified to meet emerging requirements. PMSS complies with the new 5000.1, 5000.2 and 5000.2M series of documents and the DoD standard data dictionary requirements and has built-in DoD data. PMSS was designed under Open System Architecture making it portable, interoperable and scalable

Booth #408
Digital Equipment Corporation
111 Locke Drive
Marlboro, MA 01752-1187
(508) 480-5166

Managing Knowledge Assets is the high ground of the information age. These assets include knowledge about your processes, organizations, policies, dependencies, ownership and methodologies. Properly understood and put to work, knowledge based solutions from Digital will unlock the potential of your knowledge assets and provide you with a competitive edge. In the Digital exhibit area and throughout AAAI'92 you'll see examples of services, methodologies and technologies that make managing knowledge assets possible. You'll also see several examples of how Digital integrates business, people and technology requirements into solutions ranging from business re-engineering, concurrent engineering, strategic decision support, knowledge based computer systems management, and task automation. Through sessions, demos and meetings with key Digital technical and business executives, you'll learn how to begin putting your knowledge assets to work today.

Booth #440
Elsevier Science Publishing Company, Inc.
655 Avenue of the Americas
New York, NY 10010
(212) 989-5800

Elsevier Science Publishers is one of the world's leading publishers of scientific and technical books and journals. Available at the booth are sample copies of our leading journal *Artificial Intelligence* as well as free copies of the master index of the Volumes 1–50. Other important journal titles on display include *Neurocomputing, Artificial Intelligence in Medicine, Pattern Recognition Letters, Data and Knowledge Engineering,* and *Robotics and Autonomous Systems.* Books of high technical standard are sold with a conference discount up to 20%. Visit our booth to obtain the Computer Science Catalog for a complete reference of Elsevier's publishing program in this field.

Booth #538
EXSYS Inc.
1720 Louisiana Boulevard NE, Suite 312
Albuquerque, NM 87110
(505) 256-8356

EXSYS Inc. features the EXSYS Professional Expert System Development Package, one of the most popular expert system development tools on the market. **Platforms:** MS-DOS, OS/2 UNIX, XENIX, VAX/VMS, and MS Windows, Macintosh, Sun Open Look, Motif and Presentation Manager. **Features:** Rule editor and compiler, automatic validation, custom user interface, command language, external interfaces, Hypertext, linear programming, blackboarding, frames, security, mouse interface, report generator, 5 confidence modes, Linkable Object Modules, tree diagramming of rules. **Other Interfaces:** Oracle, LINDO, NeuralWare's Professional II+ and Oil Systems PI (Plant Information System) databases and spreadsheets. **Services:** Training at EXSYS or onsite, phone technical support, prototyping, application support and special master site licenses. **Hourly demos:** Windows, Design Screens, Building Expert Systems using EXSYS, the EXSYS/NeuralWare interface, the EXSYS/PI interface. Drawings for free training.

Booth #708

Franz Inc.

1995 University Avenue
Berkeley, CA 94704
(510) 438-3600

Franz Inc. is showing a full line of programming tools for object-oriented programming with CLOS (Common Lisp Object System). Franz products are available for application development on Windows for PCs or Unix for workstations, including Sun/SPARC, DEC, HP and IBM RS/6000. Allegro CL 4.1 is a complete implementation of draft ANSI standard Common Lisp, a powerful programming language which features incremental compilation, automatic memory management, and complete debugging facilities. Allegro Composer 2.0 is an interactive development environment based on CLOS Features include class and process browsers, windowized debuggers, graphical profilers, inspector, and cross referencer. CLIM (Common Lisp Interface Manager) 1.1 is a high-level set of facilities for building graphical user interfaces for Lisp-based applications. CLIM is designed to make applications fully portable between different window systems and host environments.

Booth #638

Gensym Corporation

125 Cambridge Park Drive
Cambridge, MA 02140
(617) 547-2500

Gensym provides real-time expert system-based solutions for on-line manufacturing and process control applications. Gensym's real-time expert system, G2, and G2 Network products allow manufacturing and process engineers to intelligently describe, simulate, monitor, control and manage dynamic events for local or remote, distributed applications. Gensym has over 1000 installations of G2 worldwide in over 30 application areas ranging from closed loop control, to intelligent alarm handling to dynamic scheduling.

Booth #618

Harlequin Inc.

68H Stiles Road
Salem, NH 03079
(603) 898-1663

At AAAI'92 Harlequin will be launching a complete set of solutions for developing and delivering object-oriented applications. A new release of the object-oriented Lisp development environment, LispWorks® version 3.1, features substantially improved compiler performance. AAAI'92 will feature the first demonstration of Watson in the USA. Watson is a graphical database package for intelligence analysis. Other Harlequin products on show at AAAI'92 include KnowledgeWorks, a toolkit for building knowledge-based systems, and ScriptWorks, the world's leading PostScript RIP. There will be a preview of the upcoming release of DataWorks, a toolkit for building graphical database applications. Along with its products, Harlequin also offers a complete range of services.

Booth #728

HNC

5501 Oberlin Drive
San Diego, CA 92121
(619) 546-8877

HNC, the leading developer of neural network based solutions, is presenting in Booth #728 DataBase Mining, a production oriented model development system. It is designed for professionals with no prior neural network experience who want to rapidly build forecasting, classification and decision models from their data. A variety of other HNC products will also be shown including SkuPLAN a retail demand forecasting system that predicts sales for individual products or groups. The image processing capabilities of the Balboa 860 coprocessor will be highlighted in practical applications such as automated inspection.

Booth #608

IBM Corporation

IBM Knowledge Based Systems Marketing
1501 California Avenue, Mail Stop 6
Palo Alto, CA 94304
(415) 855-4004

The IBM booth at AAAI-92 is featuring demonstrations of several applications developed using the IBM knowledge based systems products, the Integrated Reasoning Shell (TIRS), Neural Network Utility (NNU), and IBM PROLOG. IBM developers and Business Partners will be showing the latest TIRS releases for the AIX RISC System/6000 and OS/2. Information will be available at the booth for established application developers and integrators interested in the IBM Business Partner program. A tour of IBM's Santa Teresa Laboratory programming facility is scheduled to be held Wednesday, July 15, from 7:00 to 9:00 PM. Part of the tour will include visits to 10 STL Knowledge Mining Centers™ and product demonstrations.

Booth #325

IBUKI

PO Box 1627
Los Altos, CA 94022
(415) 961-4996

IBUKI offers IBUKI Common Lisp (IBCL), a complete implementation of the Common Lisp standard running on more than 30 platforms; CONS, a specialized CASE tool for embedding/integrating Common Lisp programs into applications written in other languages; and REDUCE, a Common Lisp version of an interactive program designed for general algebraic computations.

Booth #338

ICARUS Corporation

11300 Rockville Pike
Rockville, MD 20852
(301) 881-9350

ICARUS Mentor is the easy-to-use expert system shell for the development of expert applications. With Mentor you can build expert applications in hours vs. weeks. A few keystrokes by the developer generate an expert system application for distribution that is free of any royalty charges. Features include backward and forward chaining; object-based reasoning; an interface to external programs; a procedural language; confidence factors; a configurable text editor; and a Knowledge Base Builder that creates knowledge base source files through a user dialog session. Mentor is designed to run on IBM PCs and true compatibles with DOS 3.3–5.0 with expanded memory. ICARUS Corporation is now in its 24th year of developing and marketing computer software.

Booth #546

IEEE Computer Society

10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1264
(714) 821-8380

IEEE Computer Society is the publisher of *IEEE Expert* magazine and AI related books and proceedings. As one of the most prestigious professional associations in the world, IEEE Computer Society serves its members through numerous publications, conferences and workshops. Membership information, magazines and textbooks are on display.

Booth #208

Inference Corporation

550 North Continental Boulevard
El Segundo, CA 90245
(310) 322-0200

ART-IM: Preview of new, object-oriented development environment with full-color GUI and automated functions. CBR Express: Natural language shell combines problem management/reporting with problem resolution for complete help desk automation through intelligent re-use of knowledge; accurate, rapid response; and "first call" solutions. Requires no programming; users maintain and enhance the application dynamically. SearchLite: High speed, read-only access to CBR Express case bases; provides interactive knowledge in an easily accessible, online format.

Booth #732

IntelliCorp

1975 El Camino Real
Mountain View, CA 94040-2216
(415) 965-5500

IntelliCorp, a leading supplier of application development tools and libraries, is demonstrating ProKappa and Kappa-PC, which help developers deliver applications that meet changes in their business. ProKappa is a powerful applications development system written in the C programming language for UNIX, X Window system environments and runs under OSF/Motif user interface. ProKappa uses true object-oriented programming and rule-based reasoning capabilities to provide developers with the ability to build application solutions that represent real-world problems in a natural, graphically oriented way. Kappa-PC is an application development environment for Microsoft Windows 3.1 which combines features such as object-oriented programming, a graphical development environment, and powerful rule-based reasoning to develop solutions to critical business problems.

Booth #724

Interface Computer, Inc.

One Westlake Plaza, Suite 200
1705 Capital of Texas Highway South
Austin, TX 78746
(512) 327-5344

We present our new IF/Prolog 4 1 release with exciting features: the full screen debugger for XII and OSF/Motif environments; the on-line manual for windows environments as well as for character terminals and the hypertext widget for efficient programming of hypertext applications. Third party products, based on IF/Prolog, are on display: the polymer design system EXPOD and the COBOL reengineering tool PLASMA. Interface Computer is an international group of system houses with offices in Munich, Dresden, Tokyo, Austin and Hong Kong.

Booth #346

International Association of Knowledge Engineers

11820 Parklawn Drive, Suite 302
Rockville, MD 20852-2529
(301) 231-7826

The International Association of Knowledge Engineers (IAKE) is the only international association for Professional Knowledge Engineers. Your profile as a Knowledge Engineer, a Software Engineer, a Domain Expert, a Knowledge Resource Manager, or other Applied AI Practitioner defines you as a relevant member of our society. IAKE offers Journals, Newsletters, Certification in KE, a Job Bank, a Bulletin Board, SIGS, Product Surveys, International Technology Transfer, Local Chapters, Corporate and Institutional Members, Conferences, and Seminars. IAKE is the only organization that specifically addresses matters of professional and financial concern to Knowledge Engineers and other Applied AI Practitioners.

Booth #640

John Wiley & Sons, Inc.

Convention Department
605 Third Avenue
New York, NY 10158-0012
(212) 850-6046

John Wiley & Sons offers a diverse selection of Professional, Reference and Trade books and journals of interest to those in the field of Artificial Intelligence. Stop by booth #640 to view our latest publications.

Booth #446

Jones and Bartlett Publishers, Inc.

One Exeter Plaza
Boston, MA 02116
(617) 859-3900

Jones and Bartlett Publishers presents a new publishing program of textbooks, professional reference books as well as innovative publications on new media. In addition to other recent titles, a hands-on guide to building your own robot—*Mobile Robots: From Inspiration to Implementation*, written by Anita Flynn and Joe Jones—together with some "bookbots" will be on display at our exhibit. Two robots from IS ROBOTICS, INC., THE *R-2* and *GENGHIS*, will also be demonstrated. New and forthcoming titles are available at a 20% discount at the meeting.

Booth #119

Kestrel Institute

3260 Hillview Avenue
Palo Alto, CA 94304
(415) 493-6871

Kestrel Institute is a non-profit computer science research institute focusing on formal and knowledge-based methods for incremental automation of the software process. Kestrel's research efforts are applicable to the construction of the intelligent programming environment of the future that provides automated support for all activities in the software life-cycle. Toward this goal, we carry out research on wide-spectrum languages, program transformation, synthesis of sequential and concurrent programs, and knowledge-based methods for software project management, programming environments, and life-cycle support.

Booth #747

Kluwer Academic Publishers

101 Phillip Drive
Norwell, MA 02061
(617) 871-6600

Kluwer Academic Publishers is an international publisher of scholarly books and journals in the areas of Artificial Intelligence and Computer Science. Come by to pick up sample copies of our new journals, including *AI Review, AI and Law,* and *Applied Intelligence.* Books on display include Joobbani/*Artificial Intelligence Approach to VLSI Routing*, Kowalski/*Artificial Intelligence Approach to VLSI Design*, Singh/*Artificial Intelligence Approach to Test Generation*, Brough/*Logic Programming—New Frontiers,* and Boyer/*Automated Reasoning/Essays in Honor of Woody Bledsoe.*

Booth #340

Lawrence Erlbaum Associates

LEA Associates, Inc
365 Broadway
Hillsdale, NJ 07642
(201) 666-4110

LEA continues to be at the forefront of AI and Cognitive Science. In Cognitive Science, recent titles include *Cognitive Psychology: An Overview for Cognitive Scientists* by Barsalou and *The Symbolic and Connectionist Paradigms: Closing the Gap* edited by Dinsmore. Our new series, Developments in Connectionist Theories edited by David Rumelhart has been introduced with *Neuroscience and Connectionist Theory* edited by Gluck and Rumelhart, and *Philosophy and Connectionist Theory* edited by Ramsey, Stich, and Rumelhart. Other new titles include *Modeling Creativity and Knowledge-Based Creative Design* edited by Gero and Maher, *Problem Solving in Open Worlds: A Case*

Study in Design by Hinrichs, *Text-Based Intelligent Systems* edited by Jacobs, *Human and Machine Thinking* by Johnson-Laird, *Evaluating Explanations: A Content Theory* by Leake, and *Cognitive Approaches to Automated Instruction* edited by Regian and Shute. Pick up samples of our journals, *Human-Computer Interaction* and *The Journal of the Learning Sciences*.

Booth #725

Lockheed Missiles & Space Co.

Lockheed AI Center
3251 Hanover Street
O/96-20 B/254F
Palo Alto, CA 94304-1191
(415) 354-5260

Lockheed's Artificial Intelligence Center is a corporate center of excellence with 50 professionals performing research, producing applications, and providing training in AI-based technology. The Center combines internally and externally funded research with specific system building projects, collaborating with Stanford, NASA Ames, Information Sciences Institute, and other laboratories on large scale projects. Current areas of emphasis include agile engineering, workplaces consisting of people and engineering tools interoperating in terms of shared knowledge about hardware and software artifacts; discovery, finding patterns in data using deductive database, learning, virtual environment, and parallel computation technology; and autonomy, reasoners and integrated systems for physical and computational environments in which human presence is infeasible or undesirable.

Booth #333

Lucid, Inc.

707 Laurel Street
Menlo Park, CA 94025
(415) 239-8400

Lucid is the industry's provider of Common Lisp systems for general purpose computers. Lucid also offers the Common Lisp Interface Manager, development environments, and other tools for the Lisp programmer. The Lucid Common Lisp environment offers a complete Lisp development system for work in artificial intelligence, robotics, computer vision, prototyping, scheduling and other object-oriented disciplines. Lucid also offers a complete line of C and C++ compilers and the Energize Programming System for SPARC systems. Energize offers incremental compilation and linking as well as a tightly integrated set of tools for increasing programmer productivity.

Booth #447

The MIT Press / The AAAI Press

55 Hayward Street
Cambridge, MA 02142
(617) 253-5642

Over the years, The MIT Press has built an unrivaled reputation as a publisher of outstanding books that advance our understanding of artificial intelligence, computer science and cognitive science. We continue to explore this moving frontier with prestigious authors such as Winston, Rumelhart and McClelland, Maes, Brady, and Minsky. We encourage you to visit our booth and see Churchland & Sejnowski/*The Computational Brain*, Varela & Bourgine/*Toward a Practice of Autonomous Systems*, Brachman et al./*Knowledge Representation*, Lozano-Perez/*HANDEY: A Robot Task Planner*; and from the AAAI Press, Famili et al./*Applications of AI in Manufacturing* and Balaban et al./*Understanding Music with AI*.

Booth #647

Morgan Kaufmann Publishers Inc.

2929 Campus Drive, Suite 260
San Mateo, CA 94403
(415) 578-9911

Morgan Kaufmann is a major publisher of computer books in artificial intelligence for educational and professional use. New titles at the AAAI Conference will include *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp* by Peter Norvig, *Planning and Control* by Thomas Dean and Michael Wellman, *Foundations of Genetic Algorithms* edited by Gregory Rawlins, *Advances in Neural Information Processing Systems, Volume 4* edited by John Moody, Stephen J. Hanson, and Richard P. Lippmann, *Readings in Model-Based Diagnosis* by Walter Hamscher, Johann DeKleer, and Luca Console, and *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Programs* by Bruce Buchanan and David C. Wilkins.

Booth #318

NeuralWare

Penn Center West, Building IV
Pittsburgh, PA 15276-9910
(412) 787-8222

NeuralWare offers a complete line of development tools, educational courses, and professional services for neural computing, and is expanding into application products as well. New products being introduced at AAAI include DataSculptor, the new Windows compatible data transformation tool for easy data manipulation. Several other exciting products will be on display as well, including Net-

Builder, the smart PC-package for neural computing—an embedded expert system automatically builds the network, all you do is provide the necessary data. And if you're interested in expert systems, make sure you check out the new interface between NeuralWorks Professional II/PLUS and EXSYS Professional from EXSYS. Demos of DataSculptor, the EXSYS interface, and the entire NeuralWorks product line will be conducted every hour. When you stop by, pick up a copy of *Advanced Notice,* NeuralWare's new quarterly newsletter devoted to bringing you the latest information on neural computing and related advanced technologies.

Booth #216

Neuron Data

156 University Avenue
Palo Alto, CA 94301
(415) 321-4488

Neuron Data's leading software development tools include the NEXPERT Object expert system development tool, and the Neuron Data Open Interface GUI toolkit NEXPERT Object's graphical rule- and object-based environment is available on 35+ platforms including DOS and OS/2 PCs, Macintosh, UNIX Workstations, and IBM mainframes, with complete portability across all platforms. Open Interface is an object-oriented, extensible development tool that lets you create portable user interfaces that work across all standard windowing environments. It supports native look and feel across Motif, Open Look, Presentation Manager, Microsoft Windows and the Macintosh, without changing a line of code.

Booth #439

PC AI

3131 East Thunderbird, No 8255
Phoenix, AZ 80532
(602) 971-1869

PC AI Magazine is the publication that will help you become a more productive microcomputer user and programmer PC AI brings you up-to-date information on topics such as Object Oriented Development, Expert Systems, Neural Networks, popular AI Languages and more

Booth #641

Pergamon Press

660 White Plains Road
Tarrytown, NY 10591-5153
(914) 524-9200

Pergamon Press is a leading publisher of technical books and journals, featuring *Expert Systems with Applications* and *Engineering Applications of Artificial Intelligence*. Stop by our booth for free sample copies.

Booth #718

Production Systems Technologies

5001 Baum Boulevard
Pittsburgh, PA 15213
(412) 683-4000

RAL, the Rule-extended Algorithmic Language, represents a revolutionary approach to tools for building rule-based systems. RAL is not a language or a shell, but rather a preprocessor for C. It adds OPS-like rules to the C Language in essentially the same way that C++ preprocessors add objects The OPS83 Workbench is a multi-window integrated development environment. When version 1 0 of OPS83 was released in 1983, it was the fastest rule-based language available. The enhancements incorporated into subsequent releases of OPS83 have made it even faster. Now OPS83 also offers fast development; the new OPS83 Workbench can substantially reduce the time and effort required to develop and test rule bases. PST tools are attractively priced and run on a variety of platforms, including PCs, workstations, VAXes, and MVS mainframes.

Booth #714

Quintus Corporation

2100 Geng Road
Palo Alto, CA 94303
(415) 813-3800

Quintus WorkPro™ technology, derived from Quintus' industry leading embeddable Prolog, supports rapid development and deployment of business workgroup information access and management applications. The focus is on software/hardware bug and defect tracking, customer support, helpdesk and hotline support, sales and lead tracking, and telephone messaging solutions. Deployed systems have quickly won customer's acceptance for their flexible GUI, easy-to-use Report Writer, and the ability to link to ongoing business processes The Quintus Prolog 3 1 Software Development environment, a high level rule-based application development language, is well-suited for complex systems and for traditional applications associated with daily business operations.

Booth #339

REDUCT Systems Inc.

PO Box 3570
402-4010 Pasqua Street
Regina, SK S4P 3L7
Canada
(306) 586-9408

REDUCT introduces a new form of machine learning which can handle any type of data The new product, DataLogic/R+, offers a number of unique capabilities: the ability to evaluate the quality of knowledge in your database, the ability to automatically learn from new data, the ability to inspect rules and rank them according to strength, the ability to eliminate irrelevant factors in model building which minimizes overfitting or memorizing of the data, the ability to audit decisions by tracing the rules and cases that support these decisions, the ability to handle ambiguous and imprecise data, and the ability to use rules produced in commercial expert system shells.

Booth #628

Sapiens Software Corporation

PO Box 3365
Santa Cruz, CA 95063-3365
(408) 458-1990

Sapiens Software Corporation presents Star Sapphire Common LISP, a low cost yet full-featured LISP for PCs running DOS. Star Sapphire is ideal for learning and teaching LISP as well as producing commercial applications. The product has advanced features such as 8 megabytes of workspace, CLOS, and 80% of Steele's "Common LISP, the language." It also includes EMACS, and a complete online hypertext LISP reference. Drop by the Sapiens booth and pick up a copy of Star Sapphire at the special show price of only $60—30% off list!

Booth #115

The Schwartz Associates

801 West El Camino Real, Suite 150
Mountain View, CA 94040
(415) 965-4561

Founded in 1984, The Schwartz Associates (TSA) provides consulting to vendors and users of advanced computing technologies encompassing expert & fuzzy systems, neural computing and genetic algorithms. If you are a skilled practitioner or just starting out, TSA products and services will accelerate your learning and increase technology acceptance at your company. Services include on-site-training, video courseware (*NeuroTapes, Genetic Algorithms Made Easy, Expert Systems Made Easy, Neural Network Champion Kit*, and *FuzzyTapes*), custom technology and market assessment reports, turnkey solutions, problem and product selection, project review assistance, technology deployment strategies, licensing, strategic marketing and a wide variety of published market and technology assessment reports. A contest will be held, and a videocourse will be awarded. Your participation is invited; visit our booth

Booth #247

Springer-Verlag New York, Inc.

175 Fifth Avenue
New York, NY 10010
(212) 460-1500

Celebrating 150 years of publishing, Springer-Verlag is a leading international publisher of books and journals in the areas of Artificial Intelligence and Computer Science. We publish over 1000 scientific and technical volumes yearly, serving the AI community in areas such as expert systems, knowledge engineering, neural networks, human-computer interaction, robotics, and many more. Highlights for 1992 include *Genetic Algorithms + Data Structures = Evolution Programs, Beyond Information: The Natural History of Intelligence, Artificial Neural Networks for Computer Vision, Connectionism in Context, Nonmonotonic Context-Dependent Reasonings*, and *Automatic Theorem Proving in Non-classical Logics*. See us in Booth #247.

Booth #518

Symbolics, Inc.

6 New England Tech Center
555 Virginia Road
Concord MA 01742-2722
(508) 287-1000

Symbolics provides development tools and consulting services for the solution of extremely complex application problems in industry, government, and academia. We have sold over 6000 development systems to over 1000 organizations world-wide. Our consulting group has helped major clients in the airline, telecommunications, manufacturing, utility, financial service and government areas Our featured product is Genera, the world's most powerful object-oriented development environment. At AAAI we will be introducing an exciting new product that allows Genera to be used within almost any UNIX-based site. We will also be demonstrating major new feature enhancements to CLOE, the only Common Lisp environment for the PC that supports the CLIM user interface standard; and Statice, an object-oriented database management system.

Booth #129

Talarian Corporation

444 Castro Street, Suite 140
Mountain View, CA 94041
(415) 965-8050

Talarian Corporation develops and markets the RTworks™ family of software development tools for the intelligent monitoring and control of complex, time-critical systems These powerful and innovative development tools allow customers to acquire, analyze, distribute, and display real-time data in an effective, meaningful way. RTworks combines advanced technologies, including real-time inferencing, sophisticated graphical user interfaces, real-time data acquisition, and a client-server data architecture across heterogeneous networks. This open architecture allows for easy integration with other commercial software packages such as relational databases or real-time operating kernels as well as custom programs for data collection, modeling, or simulation. Talarian Corporation provides product sales, system integration services, customer support, and training. In addition to its direct sales force, the company has a network of international distributors. The company markets its products directly to large end-users, OEMs, and systems integrators worldwide.

Booth #532

Trilogy

350 Cambridge Avenue, Suite 200
Palo Alto, CA 94306
(415) 321-5900

SalesBUILDER is a software system for building, deploying, and managing product-line configurators. It enables a sales person of customizable products to generate quotes which are accurately configured and priced. By ensuring that sales orders are technically correct before they are handed over to manufacturing, SalesBUILDER brings total quality to the point of sale. SalesBUILDER's general-purpose engine can handle the configuration of any build-to-order product, from supercomputers to modular office furniture.

Booth #127

U.S. Air Force—Materiel Command

CSTI/PIBR
Wright-Patterson Air Force Base, OH
(513) 255-7900

The Air Force Materiel Command has a unique and critical role in meeting the challenges of today's Air Force. The command provides a central focus for improving the combat effectiveness, affordability and supportability of Air Force weapon systems and processes, through partnership with Customers across all product and process lines. Broad-based expertise is applied to help customers meet the changing environment for product and process improvement. The command's mission is to accelerate the insertion of technologies for product and process improvement; provide tools, training and procedures for integrated weapon system management; support acquisition programs upon request by the acquisition executive structure; and manage selected Air Force directed programs.

Booth #224

U.S. Department of Energy

c/o Triodyne
5950 West Touhy Avenue
Niles, IL 60648
(708) 677-4730

The objective of the Robotics Technology Development Program is to develop and apply robotics technologies that will enable ER&WM operations at DEO sites to (1) reduce worker exposure and increase safety through remote operations and control of equipment (SAFER), (2) increase speed and productivity for ER&WM operations through enhanced capabilities and automation (FASTER), and (3) provide faster, more productive systems resulting in quicker completion of remediation operations that, in turn, reduce life-cycle costs (CHEAPER).

Booth #722

Venue

1549 Industrial Road
San Carlos, CA 94070
(415) 508-9672 or (800) 228-5325

Venue sells and supports tools for building information applications: MEDLEY is the complete Lisp development environment for machines across the spectrum; beyond the interpreter and compiler, you get window-based tools for coding, debugging, maintenance, and performance analysis. Our extensive library, and our large body of user-contributed software, makes application-building a snap. With LOOPS, object-oriented programming, multiple inheritance, active values, and a large library of user-interface widgets give the fastest way to build a highly-interactive application. ROOMS is the step beyond window management. Keep track of all your tasks without having to shuffle through a screenful of windows. With NOTECARDS, you can use hypertext for exploratory data analysis or argument building. Gather data, connect them together, and find a pattern, or start with a framework and fit data into it, looking for holes.

Booth #741

W.H. Freeman and Company Publishers

41 Madison Avenue
New York, NY 10010
(212) 576-9400

W.H. Freeman is exhibiting quality Computer Science Press texts in artificial intelligence, programming languages, algorithms and compiler construction. Please stop by our booth for more information.

The Robot Exhibition and Competition

Specifications & Rules

> *"This won't be a slick, polished competition. There will be a certain amount of chaos, but I can guarantee there will be a lot of excitement and enthusiasm. This sort of robotics can serve to bring together several areas of AI including those working in perception, planning, robotics, spatial and temporal reasoning, and learning. Indeed, these areas are already combining their efforts, and it seems reasonable for AAAI to facilitate and encourage this further and for the community at large to become aware of it and be inspired by it."*
>
> — Tom Dean

The AAAI robot exhibition is located in Hall 2, second level, San Jose Convention Center (enter through the exhibit area, Hall 1). As part of this exhibition, which includes robots on display and poster presentations, a robot competition is being held. This competition, which was opened to all manner of reasonably sized robots, involves a variety of events, all of which are meant to be fun and educational more than anything else.

General

This event is cast as a competition to motivate participation in the spirit of trying to develop as animate, responsive, and intelligent robot behavior as possible. The competition includes a qualifying stage involving safe roaming about the environment, followed by two stages involving spatial search, object detection/classification, and plan development and execution. Subjective parts of the scoring use an olympic-style judging. The over-arching guidelines are a variant of Asimov's laws of robotics. In order to reduce the possibility of radio or sensor interference, the competition is being conducted in two partitioned activity areas. Robots will not compete in the same activity area simultaneously. A staging area with access to the activity areas will be provided for the robot teams to prepare their robots.

There are three stages of the competition. Stage One is a qualifying stage where the robots are expected to roam about an area while not damaging people or stationary things or themselves. In Stage Two, robots are required to explore a designated area, logging any identifiable objects that they can; and Stage Three is a performance stage wherein the robots are required to find and navigate to an ordered list of objects and return. A fourth stage will be allocated for each entry group to demonstrate other aspects of their robotics work which may go beyond the competition (e.g., object retrieval, multiple robots in cooperation)

In all phases, a modified version of Asimov's three laws of robotics will be in force:

1. Within the capabilities of its dynamic range, a robot may not harm a human. (The robot doesn't have to show that it can distinguish humans from other objects. It just can't run into or over them. A robot may touch—e.g., haptic sensing—but not harm a human or other object.)
2. A robot must obey the orders given it by its operator/programmer as long as they do not violate the first law.
3. A robot must protect itself as long as it does not violate the first law; a robot shall give a warning to its operator prior to executing human orders (second law) which will violate this law.

Venue

Two polygonal areas or "rings" have been identified on the exhibit floor for the robot performances. The rings, shaped as octagons whose walls are of foamcore, stand three feet high. Each ring roughly covers a maximum area of 60 x 80 feet. A portion of the ring wall is movable, and thus has support "feet" approximately one-half inch high, two inches wide, and eight inches long. The floor is flat and made of concrete and there are no walls in the rings' interiors. Rolling tables are available near each ring for setting up computer command and control stations, including a raised platform for placing communications gear. Robots must enter the rings at the side nearest the computer stations. A staging area with access to the rings is available which will have the same wall material and floor as the rings themselves. A plan view of the rings and staging areas was provided to participants who requested it.

No navigation aids other than the floor plan has been provided. Teams may add such aids around the outside of the ring area, such as a beacon system or transponder net. No special lighting has been provided, although teams may bring their own lighting system to augment the normal exhibit hall lighting. No aids are allowed inside the rings. The judges are allowed to take points away in the "effective maneuvering among obstacles" or "autonomous exploration" or "autonomous navigation" scoring if they feel the environment has been "engineered" too much for the robot being evaluated.

Things

There are two classes of things: objects and obstacles. *Obstacles* are cardboard boxes between two and three feet high of varying widths and lengths. These will be arranged before the first stage and second stage competition into single obstacles or convex groupings of obstacles. Groupings will be setup such that there will be at least five feet separation between groups. Some number of representative obstacles are available in the robot staging area.

Objects have been constructed to be identified by the teams as follows. Up to ten object poles for each ring and five for the staging area are provided. These are six to ten feet tall, offwhite, three-inch diameter PVC pipes, with two rows of 5/8 inch diameter holes positioned at three o'clock and seven o'clock around the pipe. The poles have been constructed to stand vertically on wooden or metal bases, and the teams are expected to "rig" the poles with perceptual cues recognizable by their robot. The teams are allowed to drill extra holes to accommodate this rigging. Teams need not rig the poles if their robot cannot use special cues, or does not require them.

Note that the rigging of the poles has been accomplished prior to the objects being positioned by the competition coordinators in stage two. So unless a cue can be seen from any aspect, multiple cues per pole will probably be required. Where possible, rigging for multiple robots where the cues do not interfere is encouraged to save time.

Robots

Size

To enter the competition, a robot, including all on-board sensors and power, must be no taller than six feet and may not cover an x-y area greater than twenty square feet.

Power

The robot must be powered by batteries or tethered to standard 110 volt power supplies. Robots powered by combustion engines are not allowed.

Sensors

There is no limit on the number or types of sensors a robot may use. Use of laser sensors must be approved on a case by case basis.

Speed

During stage one, robots may not travel faster than two feet per second.

Emergency Stop

There is no restriction on robot speeds in stages two and three, but all robots must be able to be commanded to halt and remain in place within two seconds via remote control or by an on-board button or switch that can be reached by a human within one second from the ring wall closest to the robot.

The Competition

General

Because of electrical interference between robots, none of the competition stages will involve multiple robots. The first stage is considered a qualifying stage. Robots that prove not to be "safe" with regard to themselves, things, and human spectators, will not be allowed to continue in subsequent phases of the competition. Since there is a qualitative difference between the first and subsequent stages, competition points earned in the first stage will not carry over to the other stages. There will be a separate set of awards for the first stage competition and for the second and third stages combined. Only robots which meet a specific stage one qualification (see stage one rules below) will be allowed to continue on to subsequent stages.

There will be four judges at each ring for each stage of the competition. Judging for subjective scores will be done *a lá* olympic scoring (e.g., high and low thrown out, middle two averaged).

Thirty minutes will be allotted per robot performance for each stage. The first ten minutes will be used for set up and for verification of emergency stop procedures. At no time are human operators allowed in the ring area during a robot's performance.

Stage One: Robots Roaming in a Semi-Structured Environment

In this stage, the robot will roam the area of a ring, avoiding obstacles and humans (robots may touch or jostle objects or humans, but must not harm itself or the item being touched or jostled). To limit liabilities and damage to the robots, the only humans allowed in the rings during the stage one performances will be the judges. A performance will last no more than twenty minutes and must be performed autonomously by the robot. The robot may be tethered, but tether management must be performed by the robot, not the human operators. Control computers may not be positioned inside the ring. Obstacle positions will not be known prior to the performance, and the obstacles may be repositioned by the judges or by robots inadvertently in previous performances

There is no restriction on navigation techniques. The robot may "roam" via internal waypoints, trajectories, or a random walk.

Emergency stop and max speed will be verified prior to the twenty minute performance. Humans will not be allowed in the ring area until the robot has demonstrated safe operation among the obstacles for a minimum of five minutes. The teams will compete by random draw for this stage.

A robot performance begins when it starts its movement and ends when twenty minutes has elapsed.

Scoring

(60 points possible):

- *Duration* (= or > twenty minutes): 20 points. Points will be awarded at one per minute
- *Effective maneuvering among obstacles.* 20 points. This will be subjective, based on smoothness of movement, area covered, number of near calamities, and audience response.
- *Effective maneuvering among people:* 20 points. This will be subjective, based on smoothness of movement, area covered, number of near calamities, and audience response.
- *Safety-thumbs-up:* Each robot must achieve 15 points in effective maneuvering among obstacles to be able to move on to the next stage of the competetion

Stage Two: Robots Exploring in a Semi-Structured Environment

In this stage the robot will be tasked to find as many objects as possible in a twenty minute period in the ring area starting from the robot entrance to the ring. There will be up to ten objects (the actual number will be given the teams prior to the start of the competition) placed in the ring area among a random number and placement of obstacles. The objects will be numbered by the competition coordinators for use in stage three. Objects and obstacles will not be moved during a robot's performance, but they may be inadvertently jostled by the performance of previous robots. Only major jostling (as agreed upon by the judges) will be remedied in between performances.

The teams will compete in order of their performance ranking from stage one.

The performance commences when the robot starts its search procedure and ends when either twenty minutes has elapsed or when the robot has found all of the objects in the ring. The robot does not have to return to its starting location.

Find is defined as follows: The robot must distinguish between obstacles and objects, and must distinguish objects one from another in a manner that is evident to the judges, e.g., voice statement, text display on a remote computer, or a specific wagging or oscillatory motion by the robot. Further, the robot must move to within two *robot diamters* (RDs) (where the actual distance is defined by the diameter of the competing robot) of the object as part of the "find". Thus it is not enough to "see" the object from a distance.

In this stage of the competition, autonomy is part of the scoring. That is, the robots can be guided remotely to an object by a human, and/or can be told the classification and identity of an object remotely by a human, but will thus receive a lower score than robots which complete the task autonomously, all other things being equal. For example, a robot that is teleoperated but carries out a "find" autonomously can score a maximum of 80 points (see scoring below).

Note that any robot that qualifies for this stage in stage one may compete. Having a certain autonomnous capability beyond that exhibited in stage one is not required.

No humans will be allowed in the ring during the performance.

Scoring

(110 points possible):

- *Number of objects successfully found:* 7 points per object for a maximum total of 70 points.

  2 points for each object to which the robot maneuvers to within two RDs

  2 points for each object autonomously classified as such

  2 points for each object autonomously distinguished from other objects

  1 bonus point per object found will be awarded if the perceptual cues can be used by naive humans as well; e.g., instead of IR beacons or barcodes, different colored material is used
- *Autonomous exploration:* 20 points. This scoring is subjective based on the amount of human involvement in the search, as well as smoothness of motion, area covered, and so forth.
- *Effective maneuvering among obstacles:* 10 points. This will be subjective, based on smoothness of movement, area covered, number of near calamities, and audience response.
- *Inventiveness of human/robot communication:* 10 points. Subjective based on level of sophistication or cleverness of implementation.

Stage 3: Robots Carrying Out Orders in a Semi-structured Environment

In this stage the robot is directed to "find" three objects in a given order starting from the robot entrance to the ring and returning there when done. The list of objects will be given to the operator by object number, e.g., "Find objects #3, #1, and #7." where the numbers are from the numbering in stage two. The obstacles will be left in their stage two configuration, with again only major disturbances remedied between the stages and between robot performances. The objects will remain within five feet of their stage two positions.

The robots will compete in rank order of their scores from stage two. Any robot going directly from stage one to stage three will be ranked after robots which competed in stage two and in rank order with robots which competed only in stage one.

A robot performance will begin when the robot begins its search and will end when twenty minutes has elapsed or when the robot has found all of the objects and has navigated to within two RDs of the starting position.

No humans will be allowed in the ring during the performance.

Scoring

(100 points possible):

- *Task completed:* 60 points (20 points per object)

  10 points for navigating to within two RDs of each object in the order specified

  10 points for "finding" all objects in the list in the order specified
- Best completion time: First: 20 points. Second:10 points. Third: 5 points.
- Autonomous navigation: 20 points

  10 points for effectively maneuvering among obstacles. This will be subjective, based on smoothness of movement, area covered, number of near calamities, and audience response.

  10 points for efficient navigation for the task. Subjective scoring based on evidence that the robot used knowledge gained in stage two for stage three route planning and/or navigation.

Stage 4: Additional Demonstrations

These demos will take place in the rings after the stage three competition and have been coordinated with the competition organizers prior to the exhibition.

Robot Competition and Exhibition Entries

1992 Contest

Huey

Brown University Undergraduate
Artificial Intelligence (UAI) Group

Huey is, in terms of hardware, one of the simpler robots entered in the competition. The robot consists of a real world interface (RWI) B12 mobile platform and a superstructure housing on-board computers, sensors, and radio communication equipment. The RWI base is twelve inches in diameter, three wheeled, and uses a synchrodrive mechanism for steering. The robot weighs about forty pounds, is less than eighteen inches in height, and is powered by two 144 watt hour batteries. Huey's sensors consist of six sonar transducers arrayed in pairs (two forward pointing, two each right and left pointing), two additional sonars, and a pair of forward pointing infrared sensors. These, along with odometric information provided by the base, will guide the robot's behavior. There are several small on-board computers with a Motorola MC6811 handling most of the on-board coordination and communication. An Alan radio modem, operating at 9600 baud, provides communications with external processors. We are also hoping to have a flux-gate compass installed by the time of the competition.

Only the simplest of Huey's activities arise from processing actually performed on the robot. Although the MC6811 computer does participate in low-level control and obstacle avoidance when a fast reaction time is of critical importance, its principal role is to relay instructions provided by a remote Sun SparcStation running a series of modules for robot control written in C++. The radio modem connects the two; the workstation sends detailed instructions which Huey's MC6811 relays as appropriate to the base and sensors, while the MC6811 returns to the workstation whatever responses those instructions generate. This allows the robot to be guided by complicated computations, but, because of the low band width of communication, does limit Huey's speed.

Stage One

When the robot wanders, Huey's on-board computer will sample its sonars frequently to ensure that the path it is following remains clear. Meanwhile, changes in direction and responses to unforeseen obstacles (including people) will be under the control of the SparcStation.

Stage Two

In the second stage, the functions of exploration and mapping will likewise be implemented on the workstation. This will again drive the robot, over the radio link, while the on-board computer guarantees the safety of the robot's action. In mapping, the system interprets the robot's sonar readings to generate probabilistic descriptions of interesting spatial locations, and links those descriptions into a network whose arcs are described by the operations the robot must follow to get from one location to another. Dead-reckoning (and perhaps compass information) will also be applied as potential maps are generated and tested, to assist in the definition of arcs and the identification of the endpoints of arcs as previously seen distinctive locations. Huey's investigation of the circus ring thus will consist alternately of carefully probing vertices at the ends of established arcs, and of constructing the arcs out of those vertices.

Because of the limitations of Huey's sensors, the robot will only be able to identify objects by their sonar characteristics. While this should suffice to distinguish them from other aspects of the environment, Huey must rely on its human operators to provide a code number to individualize each (target) object as it is found.

Stage Three

In the final stage, Huey's operators will indicate, again by number, the objects that it must visit. The workstation will then determine a path through the graph of space that was built in the second stage that passes each of the indicated objects.

The Team

The Brown Undergraduate Artificial Intelligence Group consists, not surprisingly, of a small crew of undergraduates who thought it might be cool to play with a robot. Faculty leadership is provided by Tom Dean, and guidance, by several of Brown's graduate students in AI

ODYSSEUS

Carnegie-Mellon University

Odysseus is a small wheeled robot equipped with an arm, sonar sensors and a camera system It is connected by radio links to a pool of computers that control the robot. A main emphasis in the software implementation of Odysseus's control is the ability to act and react in real-time Odysseus combines low-level reactive mechanisms with global planning and exception handling, using a wide variety of control and AI techniques, ranging from A* planning and hidden Markov model-based speech recognition to artificial neural networks and reinforcement learning On the lowest level of behavior the robot employs several fast on-board obstacle detection and avoidance mechanisms for safely operating in unpredictable, dynamic environments. Odysseus's global navigation is map-based The sonar sensor is used for incrementally constructing a model of its environment The camera is used for detecting target objects. Odysseus is able to identify and to navigate to particular objects, as well as to explore its environment autonomously in order to gain knowledge. The robot is operated using a speaker-independent speech recognition/generation system. In addition, a graphical interface is used for monitoring the operation of the robot.

The robot is based on a HERO-2000, a wheeled robot with manipulator and a gripper There are two main sensor systems mounted on top of the robot: a black and white camera, and sonar sensors There is one fixed base-mounted sonar and one sonar sensor on the top of the robot that can be directed by a rotating mirror to give a full 360 degrees sweep. Odysseus operates tetherless: it is powered by rechargeable batteries, and connected to its front-end computers by two radio links, one for the camera, and one for a sequential RS-232 port.

Odysseus is controlled by a distributed system, consisting of 4 SUN SparcStations, one NeXT station, and a local 8088 microprocessor running BASIC. These machines, which are integrated using the Task Control Architecture (TCA), are used for interfacing and central control, map building and position control, planning, vision, speech, and fast reactive mechanisms

Navigation

Odysseus's navigation is map-based: the robot progressively constructs a two-dimensional occupancy grid map of its environment based on sonar measurements. Each vector of sonar values is translated to occupancy estimates by a pair of back-propagation networks: One network is trained to map sonar values to probabilities of occupancy, and is used as an inverse model of the sonar sensors. A second network computes the confidence in these estimates, which are used for combining multiple sonar readings. These networks have been trained before the competition to match the characteristics of Odysseus's sonar sensors

Odysseus uses a fast, anytime A* search algorithm to find a minimal-cost path through free-space, based on the occupancy grid values. The search is significantly accelerated by reusing earlier planning results (partial plans). Additional fast local search is employed to plan for exceptions in advance. The occupancy map is also used for real-time position estimation. To augment its dead-reckoning, after each step, Odysseus reestimates its position with respect to the global map. This is done by building a local map from the most recent sonar input (using the two artificial neural networks described above), and then optimizing the match between this local map and the global map using gradient descent search. We have found that this procedure successfully deals with cumulative errors in dead-reckoning.

Using the TCA, map building, planning and position estimation is all done in real-time, concurrently with plan execution.

Odysseus uses two techniques for obstacle detection and avoidance. While the robot moves, it continually monitors its sonar sensor, and slows down and/or stops if something blocks the way. The robot also monitors its wheels encoders to detect stuck or slipping wheels. This "guarded move" is implemented on the robot platform itself, to make it more reactive and independent of radio link delays. In addition, a fast closed-loop controller is employed to navigate around unexpected obstacles and avoid collisions by modifying the direction of the robot. In the current implementation, this controller is realized using a back-propagation network trained with a reinforcement learning algorithm.

Vision

Vision is used primarily for finding and identifying distinguished objects. The objects are marked by a circular pattern, divided vertically into two regions: a "checkerboard" region of black and white squares, which is used to distinguish the object, and an individual name region, which consists of horizontal lines that uniquely identify that object. The camera system periodically takes pictures and looks for objects in the scene. The information extracted from the picture (range, relative direction, identification) is used to construct a two-dimensional target map similar to the occupancy maps built from sonar readings. The map keeps track of the position of detected marked objects as well as known object-free regions in the operation area.

Exploration

Exploration is the process of maximizing knowledge gain. Odysseus evaluates the expected knowledge gain along a path using inverse models of sonar and camera sensors. Odysseus's exploration control consists of two components. Exploration is combined with general navigation in a way that optimizes knowledge gain while the robot is in motion. Consequently, the path planner can take the expected knowledge gain into account. In addition, the path planner can be used for pure exploration, operating without a specific goal position. Then the robot navigates to places where the expected knowledge gain is optimal. Pure exploration is employed to hunt for objects.

In the current implementation, exploitation dominates exploration: Whenever Odysseus knows about some objects that it has not yet been to, it moves towards the nearest one. Here exploration plays a minor role. Once all detected objects have been approached, the robot explores to find new objects.

User Interface

Odysseus's user interface handles oral and graphical commands. A speech recognition and generation system is used to command and, if necessary, teleoperate the robot. The Sphinx speech recognition, developed at CMU, performs speaker-independent speech recognition in real-time on arbitrary pre-defined grammars. Oral speech commands include low-level and high-level commands for specifying tasks. Odysseus has a speech synthesizer on board, which allows the robot to give feedback and carry on a dialogue with humans.

Odysseus also has a graphical interface, which is used mainly for monitoring the current state of the robot (its maps and plans) as well as the state of the control programs. Odysseus can also be controlled and teleoperated with this graphical interface, although the system usually operates totally autonomously

Buzz

The Georgia Institute of Technology Denning Mobile Robotics Entry.

Buzz, a MRV3 robot provided by Denning Mobile Robotics, is a three wheeled holonomic robot twenty-seven inches in diameter and thirty-six inches high. Buzz has twenty-four ultrasonic sensors mounted in a ring twenty-two inches above the floor. A 68000 microprocessor running OS/9 acts as the robot interface, controlling other microprocessors in charge of the motors, ultrasonics and a IR beaconing system. High level commands such as "turn 180 degrees" or "move forward at one foot per second" are transmitted to the robot over a 9600 baud serial link. The IR beacon system provided by Denning mounts on the head of the robot. Serial numbered beacons will be placed on the goal tubes to allow discrimination between the goals. The beacons are rated to be visible from forty-six feet and the detector returns azimuth and elevation to detected beacons

We plan to use two different methods of controlling Buzz. Our primary system is a Sparc IPC computer with a digitizer board.

Both video and serial data are transmitted via RF datalinks. A new experimental system uses an on-board transputer multiprocessor with up to sixteen T800 transputers and a video digitizer. The Sparc system is programmed in C and the transputer in OCCAM.

Perception

We use a ring of twenty-four ultrasonic distance sensors for obstacle avoidance, and shaft encoders to determine robot position. One black and white CCD video camera will be used to find the objects. A small light source will be mounted near the camera. For object identification, we will attach a simple target cue on each object, visible from all directions. We are also planning on using infra-red beacons in stages two and three.

Software Architecture

Buzz employs a schema-based reactive control system to make its way around. This control system is a subset of the hydrid hierarchical/reactive autonomous robot architecture (AuRA). The general navigation and movement task has been broken into low-level low-cost independent processes (at present count—fifteen). There is no explicit hierarchy of processes, but they can be grouped under the headings of "planning," "motor control," and "perception." In general, motor control processes reference "raw" data gathered by perceptual processes to generate movement vectors. These vectors are summed to yield resultant movement commands. The planning processes select which motor and perceptual processes should be active and what parameter values they should use.

Stage One

The wandering behavior will be accomplished with a small suite of active motor and perceptual processes. The ultrasonic range sensors will be used to avoid obstacles and people. A novel local memory process will ensure complete coverage of the arena.

Stage Two

The object-finding task requires a more complex suite of motor and perceptual processes. We will employ a "search, move, stop, search" strategy. The vision system will search for objects when the robot stops. Then a movement command will be issued, and the ultrasonic sensors will be used for obstacle avoidance.

Stage Three

World knowledge gained in the earlier stage will be used to move to specific parts of the arena.

Advanced Capabilities

We intend to demonstrate the transputer system during at least the phase one portion of the competition. We also plan to demonstrate Buzz's ability to navigate in complex, unmapped environments. A new motor schema approach has been developed which allows navigation around box canyons, and similar obstacles. Buzz has access to the sensor fusion effects (SFX) architecture for integrating observations from multiple sensors into a single percept, which is expected to be used to demonstrate advanced perceptual abilities. The SFX architecture uses a state-based control scheme to provide three levels of feedback from a perceptual process to its active sensors. Experiments in other domains have shown how perceptual processes under SFX can recalibrate errant sensors or ignore random spurious observations, resulting in a higher confidence in the percept and robust performance.

The Team

The Georgia Institute of Technology team is made up of ten students, both undergraduate and graduate, with one faculty advisor.

TJ

The IBM AI Mobile Robot Team

TJ is built on a RWI B12 three-wheeled omnidirectional synchrodrive mobile platform. The robot is twelve inches in diameter, stands three feet high, and weighs in at a svelte fifty-five pounds. TJ has an array of twelve short range infrared proximity sensors and four longer-range sensors to aid in local navigation. For added reliability with respect to collision avoidance there are also four forward looking Polaroid sonar ranging devices and four side-looking units. Object detection is accomplished through the use of a planar rotating infrared phase-based range sensor. Object detection is achieved using a small video camera connected to an on-board low-bandwidth vision system which subsamples the image for analysis. Object identities are announced by a speech synthesizer. Local navigation and reflexive goal-seeking behaviors are controlled by an on-board network of eight microprocessors connected by a high-speed multi-drop serial network. These run a distributed behavior-based subsumption controller which reevaluates the robot's situation every seventy milliseconds. All these systems are self-contained and powered by an on-board battery system. Supervisory control and a symbol-based human interface is provided by an off-board IBM PS/2 Model 70 workstation connected to the robot over a bi-directional 915 MHz 9600 baud spread-spectrum Arlan radio link

Stage One

Wandering as well as person and obstacle avoidance is accomplished completely autonomously by the on-board subsumption control system. The robot goes forward whenever possible but swerves if the IRs or sonars detect a nearby obstacle. The robot also has progress monitoring to keep it from getting stuck in corners and stall sensing to respond to unanticipated collisions.

Stage Two

In this stage the off-board workstation sets a number of odometric goals for the robot so that it progressively covers the entire ring. Local navigation and collision avoidance is performed using the onboard processor network as in stage one. After traveling a specific distance (about ten feet), whether or not the specified goal is reached, the robot stops and examines the environment with its range scanner. A coarse angular resolution is used to detect the presence of potential objects (the posts). The workstation then directs the robot to take more detailed scans in particular areas of interest to verify the presence of an object and to better localize it in range and angle. When multiple known objects are re-sighted, the robot uses the observed configuration to recalibrate its odometry.

When instead a new object is detected, its position is recorded relative to the robot's current odometrically determined position. The supervisory program then directs the robot to approach the perceived position of the object. We intend to mark the objects using a wrap-around pattern of black and white horizontal stripes to encode a four bit binary number. When the robot has approached close enough, the on-board camera snaps a picture and associated circuitry averages and subsamples the analog signal then digitizes a vertical strip of the image. This is sent over the radio link to the workstation which interprets the code and in return sends a message to the on-board speech synthesizer to announce the object's identity After this the search pattern continues as before.

Stage Three

For this part the robot will be started off in the same position and orientation as was used in stage two. The desired object is specified either by typing on the console of the workstation or by showing the robot's camera the desired stripe pattern. This time the supervisory program plans a rough path to the object desired and uses this to guide the robot's wandering. As before, the robot stops occasionally to recalibrate itself. When an object is found in approximately the right place, the robot approaches and views it with the camera. If it is the object specified, the robot stops and announces that the goal has been found. Otherwise is looks nearby for other objects that might be the desired one. After the first object is found, the operator specifies the next target and the robot goes after that one in a similar fashion

Stage Four

TJ has a few other tricks that are not currently integrated into its exhibition program. One is a demonstration of real-time on-board autonomous object tracking use low-bandwidth vision to detect and follow black pants legs against a white floor. Another demo involves learning how to follow walls using a hard-wired reward function and reinforcement learning. There may be other demos as well, depending on what is ready at the time of the conference.

The Team

The exhibition robot was built and programmed by a Ph.D. researcher, with the assistance of a BS level technician. The IBM robotics group has an additional three members who have been working on different projects.

Scarecrow

David Miller and Jacob Milstein

Scarecrow is a completely self-contained totally autonomous mobile robot that was specifically designed and built for the 1992 AAAI mobile robot exhibition. The robot is approximately four feet tall, roughly cylindrical in shape, with a diameter of approximately two and one half feet. The robot moves using differentially driven wheels, and a set of casters. All batteries and motors are located within six inches of the ground to keep the robot stable. The robot is equipped with tilt sensors to disable the actuators should the robot start to tip for some reason. The robot has a programmable top speed that can be adjusted to be less than two feet per second. The robot is ringed with soft bump sensors along its lower base. The robot has a sensor ring on its top which can read a conductive bar code of any object which is labeled and with which it comes in contact. The processor and control electronics were all custom designed for this robot, and are extremely power efficient.

Scarecrow is a very reactive robot. It is being built, in part, to demonstrate the capabilities and competence that can be accomplished by using a strictly reactive architecture for well defined tasks such as this exhibition.

The robot initially heads off in a random direction until it senses an obstacle. Objects and obstacles can be distinguished by height and labeling. All objects will be labeled with a numbered bar code. Scarecrow can sense and read this bar code when it comes in contact with the object. When it sees the label it will report that it has seen an object and display the object's identification code.

Whether it has encountered and object or an obstacle, Scarecrow will proceed part way along the perimeter of the obstacle, and then head off to explore a new region of the arena. Obstacle detection and avoidance are accomplished by reactive routines that connect the touch sensors to the two drive motors. Contact on the left side of the robot affects the speed of the right drive allowing the robot to turn towards or away from the thing detected. If it is an obstacle, the robot moves away. If it is an object, the robot moves in so that it may read the bar code. Scarecrow maintains almost no state information. For it to visit the objects in the correct order, it is programmed by the operator with object ID'S in the desired order. It uses a simple FSA to keep track of which object it has last seen, and which it wants to see next. If it comes across the correct object, it recognizes it and announces it has found it. If it comes across the wrong object, it treats it like an obstacle, and continues to look for the next object.

People have been working on general purpose robotics for forty years, with very limited success. We prefer to think of robotics systems in a similar vein to expert systems. A robot should be designed for a specific task or set of tasks When the scope of the robot is limited, reactive behavior control techniques have been quite successful. This is especially true when the hardware is developed in concert with the software. Scarecrow pushes this model of robotics to the limit. Scarecrow is specifically designed for this one particular contest, and is good for little else other than an educational example. As an educational tool, Scarecrow is excellent. This is because the reactive behaviors are incredibly simple. Scarecrow is a Braitenburg Vehicle with a mission.

The Team

Scarecrow is not officially sponsored by any organization. Team members include David P. Miller from the MIT Artificial Intelligence Laboratory (on sabbatical from the Jet Propulsion Laboratory / California Institute of Technology) and Jacob Milstein from the Leslie-Ellis School in Arlington, Massachusetts.

Uncle Bob

The MITRE Corporation

Our robot and its accessories Denning MRV with a ring of twenty-four sonars mounted about 0.7 meters from the floor with fifteen-degree separation. six sonars are mounted about 0.1 meters from the floor more or less evenly around the robot with baffles to increase the dispersion angle of the sonars. Pitch and roll sensors, as well as a laser target reading system on loan from Denning. The robot's onboard processing includes a 68000 which is used to control the base's actuators and read its sensors (excluding the laser target reader). The robot intelligence is located in an Apple Quadra which is onboard powered by a 100 amp hour DC source. The Quadra has additional RS232 ports. The Quadra is connected to the base via RS232 and to the laser target system via RS232. In addition, the Quadra communicates to the command and control (C&C) computer, located off board, via an RS232 RF link operating at 9600 baud.

The C&C computer (a Macintosh IIFX) is used to display robotic telemetry and to start and stop the robot's autonomous activity. Uncle Bob is tetherless.

In general the robot is coded using the REX/GAPPS programming environment which exists inside of a lisp system.. Behaviors are coded REX and then sequenced with reaction plans coded in GAPPS. The compilation results is a definition of a circuit which implements in constant time cycles the various algorithms which are used to control the robot. The circuit definition is then translated into a simulation of the circuit in C. This C-circuit is then linked with the runtime environment, also coded in C, which performs the communications with the robot, sensors, and the command and control computer.

All navigation on the robot is orchestrated through the use of navigation templates (or NaTs). NaTs are similar to potential fields, except for the addition of knowledge about the goals which allow for decisions as to clockwise and counterclockwise motions the robot must take around obstacles with respect to the goal in order to satisfy the NaTigation Plan.

Stage One

Stage one is handled by a behavior that causes the robot to wander around its environment, maintaining a safe distance between it and the nearest obstacle, as seen by the sonars. If the robot has chosen a way point which it cannot safely reach, the point is abandoned and a new point is chosen. To keep the robot from looping it will keep a list of those places that it has been and pick a direction which will move it to a new location (if possible). NaTigation plans support obstacle avoidance during wandering, but low-level runaway behaviors take over when objects appear suddenly at too close a range.

Stage Two

In stage two, target identification will be accomplished through the use of the Laser Target reading system, which uses a spinning laser and laser detector to find coded patterns of reflectance in its environment. Using this system the robot will locate targets, actively triangulate on the target's relative position and move to the target. Mapping software will record the environmental characteristics at a target location so that precession can be eliminated if the robot visits that location again. As the robot leaves a location a link in a global map is formed which allows the robot to remember where the next target is relative to the last position.

Stage Three

Using the relative map acquired during stage two, the robot will plan a sequence of targets to visit in order to satisfy the request for this stage of the competition. At each location the robot will alleviate precession error, using the stored local model and then move to the next target in the sequence or an intermediate target by accessing the relative position of the next target stored in the graph. If the robot must move to a target which is not attainable in its stored map of the environment it will move into an exploratory mode, wandering until such time as the time limit has been exceeded or the target is found. During this exploration it will be actively adding to its map so that if the target is found it will have information which links the target with its previously acquired global knowledge.

The Team

Our team is composed of Marc Slack and Pete Bonasso of the MITRE Corporation.

Soda-Pup

NASA-JSC-Automation & Robotics Division

Soda-Pup consists of a mobile robotic platform with on-board power supply and processing as well as off-board OS/2 and UNIX based computers that provide the primary processing power and serve as the operator interface. The off-board computers communicate via TCP/IP, and a 9600 baud radio link provides the serial communications pathway between the main on-board processor and the off-board computers. The Soda-Pup robot uses as its mobile base a Nomad 200 mobile robotic system built by NOMADIC Technologies, Inc.. The Nomad 200 has a cylindrical structure, consisting of a lower base and an upper turret. The base structure houses the electrical power distribution system and the drive system. The drive system provides synchronous translational and rotational motion to Nomad's three wheels as well as independent rotation to the turret. The turret functions as the superstructure of the mobile robot, housing both the onboard computer systems and the majority of the sensors.

Soda-Pup's maximum diameter is twenty-one inches, and its maximum height is four feet. The base of the mobile robotic system houses a 432 Watt-Hour rechargeable battery pack The robot contains twenty interleaved, concentrically distributed binary pressure sensors for detecting contact with the robot's bumper; an encoder for odometric dead-reckoning position and orientation determination; sixteen fixed time-of-flight ultrasonic sensors, concentrically distributed around the top portion of the turret, for detection of obstacles within the range of 17 to 255 inches, and fixed infrared-reflectance sensors, concentrically distributed around the bottom portion of the turret, for detection of obstacles within the range of zero to twenty-four inches.

A color CCD camera is mounted on top of the turret for object detection. An infrared laser—CCD camera ranging system, mounted on top of the turret, is included for position estimation of objects within the range of 18 to 120 inches. The structured light is produced by a ten MW, 685 nm infrared laser.

Soda-Pup's maximum translational speed is eighteen inches per second. Emergency stop is facilitated by a 1.5 inch diameter red button mounted on top of the turret. This button physically switches off power to the drive system. The button locks in the "power off" position when pressed. A graphical emergency stop button will also be provided on the operator interface display.

The Soda-Pup robot architecture is built upon individual processes (running on a mixture of OS/2 and UNIX platforms), which communicate with each other using an in-house developed interprocess communications package. By organization of these processes into six major functional modules (sense, perception, knowledge, motivation, planning, and action), data progresses from low-level data acquisition processes through higher-level data interpretation functions, and eventually to the planning and action processes which plan for and execute selected behaviors The sense module is responsible for acquiring and performing preliminary processing of the sensor data before distributing it to the perception module. The action module, at the other end of the data flow, transforms task plans into actual motor commands.

Perception

The Soda-Pup's perception module currently consists of three processes: imminent collision detection, obstacle detection, and object identification. The collision detection process quickly detects obstacles that pose an immediate threat by entering a "danger zone" around the robot. The obstacle detection process detects, over time, obstacles in the robot's environment. The object identification module detects and identifies predefined, colored objects

The imminent collision detection process serves to quickly detect objects that threaten to collide with the robot. An object is considered threatening when it enters a configurable "danger zone" about the robot. The collision detection process uses three sets of sonar values (taken over consecutive time steps), a single set of infrared values, and a single set of tactile information from the robot's bumpers If there is a cluster of sonar values within the danger zone surrounding the robot, or an infrared value within the zone, or if there is a bumper hit, a vector indicating the relative position of the threatening object is passed to the reactive component of the planning module for quick avoidance. Clusters of sonar values are used to avoid reacting to bogus sonar data. Clusters are not used for infrared or bumpers because the ranges of these sensors are small enough that immediate reaction may be required.

The obstacle detection process, active in all three stages, creates a certainty map of "filled" space using sonar and infrared sensor data. The certainty map divides the robot's environment into a grid of one inch squares. A sensor "hit" is represented on the grid as a positive certainty that the space about the hit is filled by some object. The distribution of certainty is Gaussian, with the peak at the actual value of the hit. The space between the robot and the hit is given negative certainty. The certainty is more negative closer to the robot and goes to zero linearly. New certainties are combined with old certainty values already on the grid using standard certainty management equations. Obstacles are determined by finding zero crossings in the grid boundaries between positive and negative certainty. These boundaries are smoothed into line segments and sent to the knowledge module where a line-segment global map of the world is maintained.

The object identification process is used in stages two and three to detect color-coded objects placed on the PVC poles. A color CCD camera mounted on the turret will constantly transmit images to the off-board computer in NTSC format. Images will be captured in a frame buffer and translated into HSV color format. The hue image is then segmented by multiple thresholding operations into eight color regions which are then divided into blobs and filtered by size. Geometric relations between the blobs are then determined. These are compared to a model of the object. If the relations match the model closely enough, an object is said to be detected. The arrangement of colors is checked to uniquely identify it. Range determination will either be done based upon image size, or by using an active laser-ranging system. The robot will indicate via the operator interface the location and identification of detected objects.

Knowledge

The primary responsibility of the knowledge module processes are to build and maintain global maps of the world for use in task and motion planning. Three global maps are maintained: an obstacle map, a known region map, and an object map. The obstacle map is used to plan around previously discovered obstacles in all stages of the competition. The known region map is used for exploratory behavior in both stages one and two to direct Soda-Pup towards regions not yet traversed. Finally, the global object map maintains the location and identity of objects detected in the environment during stage two. This information is used by the planning module to formulate an ordered, directed search of the desired objects (as specified by an operator via the user interface) during stage three.

The obstacle map is represented as a list of connected points (forming line segments) which indicate physical barriers in the environment as detected by perception module processes. Map maintenance consists of incorporating newly received data from perception into the global map. Currently, a simple brute force method of removing old data and inserting new data is used, although more powerful approaches using line-matching techniques for correction of odometric errors will be incorporated into Soda-Pup if time permits.

The known region map also consists of a list of connected points. At regular time intervals, based upon the robot's position and currently "visible" obstacles, a local known region is computed by starting with the surrounding "horizon" region (determined by the maximum reliable range of the sonar sensors) and then algorithmically removing areas that cannot be "seen" due to obstacles. The area inside this irregular-shaped, polygonal region represents the robot's local known region during that particular time interval. The local region is then merged into the global known region map. The global known region map represents the union of all previously known regions. Other polygonal manipulation routines map unknown regions that can form within the surrounding global known region.

Motivation

The motivation module serves as Soda-Pup's high level "behavior selection system." The motivation module is implemented as a neural network with, in its current version, fixed weights. The basic components of the network consist of sensory nodes, arousal nodes, and behavior nodes, denoted, respectively, as S, A, and B nodes. Each S node, when active, simply represents the current existence of some condition in the world (as determined by perception and knowledge processes). Each B node represents a behavior that Soda-Pup may elicit. The set of B nodes constitute Soda-Pup's entire repertoire of behaviors. The "firing" of a B node causes the planning module to formulate a task plan appropriate for eliciting the specified behavior. Weighted pathways from S to B (S->B) provide the mechanism with which sensory events direct the behavior. The A (arousal) nodes are needed to "energize" behavior. Without an active arousal source, no behavior can be elicited. Without this additional influence of the A nodes, the robot acts as a purely reactive agent where behavior is determined solely on the basis of currently perceived external conditions. The three A nodes used in Soda-Pup represent different sources of arousal, namely "fear," "hunger," and the "need to obey operator-issued commands." These nodes serve to mediate the influence of environmental stimuli by taking into account the internal "needs" of the robot (such as power level). Only S nodes that are also compatible with the current arousal source can become active and elicit behavior. Because multiply active arousal sources can result in conflicting behaviors, the A nodes compete with each other for activation in a winner-take-all scheme, such that only one A node is active at any given time. By adding a bias to the A nodes, Soda-Pup becomes more predisposed towards A nodes with high bias values than A nodes with lower bias values. "Fear," "obey," and "hunger" arousal nodes are assigned descending bias values. This predisposition maintains Soda-Pup's adherence to the modified version of Asimov's laws of robotics.

Planning

The planning module is responsible for converting a specified behavior (as selected by the motivation module) into "concrete" action commands for body, arm (in future versions), and sensor positioning and control, along with activation of appropriate perception module processing states. The planning module provides long term task plans for achieving desired body locations as well as for proper sequencing of movement with perception and knowledge module activities. Planning also generates the specific global motion path plan for the body to achieve a desired location, and for the positioning of movable sensors as required by the perception processes. Localized reactive path planning is also generated during imminent collision conditions.

The task planning process is implemented as a behavior sequencer that builds task stages from lower level behaviors. This provides a hierarchy of behaviors that can be arranged as required to perform the requested function. An individual behavior can be as simple as a "canned" script of actions, or it may involve complex interactions dependent upon perceptual and knowledge processes (such as searching for an object in a known location), but the action steps are always sequences of low level functions that can be strung together. The task planner formulates action steps for the following behaviors: stay, operator move, collision avoidance, go home, wander aimlessly, move to known, explore unknown, revalidate the known, identify objects, and search for objects.

At the time this abstract was written, body motion planning was leaning either towards a vector based approach, one of several variations of the potential field approach, or a 2-D configuration space approach. Since the robot is circular in shape, more advanced path planning methods that could handle non-zero turning radius or articulated robots were not deemed necessary.

The reactive imminent collision motion planning is based on a potential field approach. This allows the robot to dodge around moving objects entering the perceived "danger zone" while still maintaining, as much as possible, the general desired heading.

The Team

The Soda-Pup team consists of seven people (two MS. and three BS. level NASA/Johnson Space Center employees), and two BS level Lockheed Engineering and Sciences Company employees). Out of this team, four are primarily responsible for Soda-Pup "intelligent" software development, two for maintenance of a robot simulator, and one for inter-process communications software development.

HOMER/BugEyes

San Francisco Robotics Society and the Palo Alto Homebrew Robotics Club

HOMER is an original, conceived as a proof-of-concept automatic vacuum cleaner. It has an integrated front-mounted, retractable beater-brush, a suction unit, and a replaceable filter/collection bag. HOMER is relatively compact. It is approximately 530 mm long, 430 mm wide, and 430 mm high. It weighs about 16 kg. The base consists of two large (300 mm diameter) mid-front mounted drive wheels with nine degrees of negative camber, and two small (95 mm diameter) castoring wheels at the rear corners. The base can be considered holonomic for purposes of path planning. Chassis construction is monocoque, with a composite of aircraft-grade plywood and fiberglass–epoxy.

The primary control computer is an on-board NEC V40-based Ampro board with 512K of RAM and an on-board 1.44 mb floppy disk drive. The computer is programmed in C++, and uses an Intel 8255 peripheral controller to generate a custom 40-bit bus for interface to the motor controllers and the sensors. Propulsion motor control uses two Hewlett-Packard HCTL-1000 motor controllers providing proportional PWM control to two H-bridge motor drivers. An infrared remote-controller board and custom sensor-processing boards also communicate through the bus. Power is provided by two independent twelve volt, five amp-hour Ni-Cad battery packs One battery provides power to the electronics, the other to the drive and vacuum motors. The electronics-motor interfaces use optical couplers for complete isolation. All circuits are protected with circuit breakers or fuses

A small hand-held pushbutton controller, similar to a television remote control, allows remote activation, mode-switching, and emergency stop, with a control range of approximately ten meters. The remote unit also allows a limited degree of teleoperation. Two large exposed and lighted pushbuttons are mounted on the exterior of the robot to allow easy manual shutdown if necessary. A flashing yellow strobe light is mounted on top and activates when the motor circuits are turned on.

Landmark identification and some collision detection will be provided using a ring of twelve Polaroid ultrasonic sensors and twenty-four photocells. These are installed, but are not currently operational. In addition, two forward-looking ultrasonic sensors and two reflective infrared sensors provide forward obstacle detection. The entire robot is surrounded with a "floating ring" bumper about 100 mm high. The bumper has about 10 mm of compliance in all directions, and its movement can activate any of eight microswitches designed to detect the direction of bumper displacement. The vacuum beater-brush can extend through the front of the bumper ring, and is equipped with a separate two-piece bumper.

Vision

The robot base will be carrying a stand-along object recognition system called "Bugeyes," a hardware neural-network system modeled roughly after the assumed operation of complex insect eyes. In its present state of development, BugEyes can distinguish between a specific object and any other objects. The system is implemented on a single six inch by eight inch circuit board. It uses a Texas Instruments TC211 small image area CCD camera, a Dallas Semiconductor DS5000 8-bit microprocessor running at 16 MHz, and an additional 128K data area RAM.

Operation

The system has three modes of operation. In the learn mode, the system begins building a database about an object being imaged by the camera. During learning, the robot points the camera toward the single object to be recognized. The object must be imaged at all perspectives from which it will need to be identified. In the forget mode the robot points the camera at other objects that are not to be recognized, and the system selectively erodes the database. Finally, in recognize mode, the vision system will indicate whether it is looking at the learned object or at other objects. Currently the learn and forget modes take twenty to sixty minutes each, depending upon circumstances. However, recognition takes only a fraction of a second.

The Competition

Our stage one attempt will be a random walk with infrared sensors and the bumper ring detecting objects and the robot responding accordingly. It is not expected that the dead-reckoning information provided by the wheel motor-encoders will allow the robot to build a reliable internal map of the unstructured space.

In stage two, we will use the BugEyes vision system during a second random walk. The system can identify a broad range of real-world objects, and we have yet to determine what, if any, modifications might need to be done to the competition objects. Because the BugEyes system is a yes/no single-object recognizer, we do not plan to participate in stage three. In stage four, we intend to demonstrate the unique characteristics and flexibility of the vision system and, possibly, the vacuum-cleaning capabilities of the robot.

The Team

Brad Smallridge, San Francisco Robotics Society; Art Gaffin, Object Recognition Specialist; Roger Gilbertson, Mondo-Tronics; Frank Jenkins, JRL Consulting; and Richard Frather, Palo Alto Homebrew Robotics Club.

FLAKEY

SRI International

FLAKEY is a custom-built octagonal robot approximately two and a half feet in diameter and three feet high, weighing 300 pounds. Locomotion is by two independently-controlled wheels located on the sides; maximum speed is about five feet per second. Sensors include a bottom-mounted ring of touch sensors, a circular array of twelve Polaroid sonar sensors, and a structured-light system using an infrared laser and CCD camera. Internal computers include microprocessors for controlling the wheel motors and sonars, and a SUN-3 master that coordinates the other controllers, collects and process the structured light, and communicates with an off-board controller via a two hundred kb wireless ethernet bridge. Offboard processing has been done on a number of processors, including CM-2, IRIS, and Sun computers. This demonstration will use a Sun SparcStation. There is also an implementation of a simulator for FLAKEY that runs as a process on any SparcStation. The simulator does not have the structured light sensor.

The controller is written in Lisp with C subroutines where necessary for speed. Multiple real-time processes are implemented as a software round-robin queue. Cycle time for the queue is targeted at one hundred ms. There are processes for basic communications and screen display, motion control, and sensor interpretation, and mapping.

A fuzzy control system directs FLAKEY's motion. Fuzzy control rules consist of an antecedent determining the strength of applicability of the rule, and a consequent stating its intended action. Control rules can use interpreted sensor inputs as well as prior information in determining their output.

Behaviors are implemented as sets of control rules with optional spatial control points. Sets of control rules define acceptable motions in achieving the control point, or in moving without a precise spatial goal. A behavior's set of control rules must implement goals that can be interpolated to achieve optimal control. It makes no sense to have a single behavior with competing rules to go right or left around an obstacle, since these goals cannot be interpolated. Behaviors are modular and can be combined to achieve simultaneous goals.

Sequences of control points with associated rules between the points can implement sophisticated behaviors. Control rules a flexible and easily-debugged motion programming language. Supervisory processes can control FLAKEY by switching behaviors at appropriate moments.

FLAKEY's sensor interpretation routines are directed towards producing a locally consistent metric perceptual space. Because the sensors are short-range (one–two meters) and the wheel odometry is unreliable over longer distances, no attempt is made to keep a consistent metric global map. Instead, local perceptual information is stored in a topological global structure.

There are two types of representations in the local metric space: area-based and surface-based. Area-based representations are similar to the occupancy-grid model: sensor readings are collected with little interpretation to indicate open and occupied areas. Area-based information is used to do local obstacle avoidance.

More complicated motion and planning requires surface-based representations, in which sensor readings are interpreted to extract surface patches and other significant perceptual features, and these are grouped into coherent configurations and objects.

Currently we have implemented the lowest level of the surface-based representation, extracting coherent surface patches from the sonars and structured-light sensors, and performing sensor fusion and model updating in real time. This process gives a fairly detailed surface model of the local environment, and keeps FLAKEY registered with respect to it. There is a limited motion recognition capability in the structured-light interpretation, which can be used to distinguish moving objects from a fixed background, and track them. We are working on the grouping and abstraction routines, and hope to be able to demonstrate them.

The surface information in local perceptual maps is stored in a structure that reflects the topological structure of the area that FLAKEY has visited. The map contains local metric information as well, but this information is not combined to form a precise global metric map. The map is hierarchical structured, with larger structures at the more abstract levels of the hierarchy, and local metric information at the bottom.

Stage One

Six control rules implement a wandering behavior that keeps FLAKEY away from obstacles and competitors. A supervisory process will switch in "unsticking" behaviors if it stays too long in a given area.

Stage Two

A simple planner will control behaviors to force FLAKEY to move towards unexplored areas. Along the way, FLAKEY will build up a surface-based local maps of the environment, combining these into a global map that it uses, along with imprecise wheel odometry, to keep track of where it is. Objects to be identified are recognized with the structured-light sensor and their locations stored in the global map.

Stage Three

The strategy here is similar to that for Stage two, except the planner will generate goals to move to each of the objects.

The Team

Team members consist of Nicolas Helft, Kurt Konolige, Karen Myers, and Alessandro Saffiotti (International Fellow).

Chip

University of Chicago

Chip is roughly cylindrical with a diameter of about eighteen inches and a height of about three feet. It rides on a synchro-drive base from RWI and is equipped with a bumper near floor level to detect collisions. The robot has eight sonar range sensors good to about twelve feet and sixteen I/R range sensors good to about three feet arranged around the robot eighteen inches above the floor. The robot is also equipped with a simple Hero II arm which can reach the floor. Sitting on top of the robot is a color camera on a custom pan/tilt head. The image from this camera is broadcast off the robot via radio to a Sun SparcStation equipped with several DataCube image processing boards. All motors and sensors on the robot are managed by on-board microcontrollers that communicate with a Macintosh computer off the robot using a 9600 baud radio modem. Robot control is spread across the on-board microcontrollers, the Macintosh, and the SparcStation.

Stage One

Chip will use a potential field based approach to move from place to place in the competition area. The places will be chosen by a very simple planner to ensure that Chip does not spend too much time in any given area. Should obstacles appear suddenly close to the robot low level routines will stop or move the robot away. Sonar and IR data will be used by the potential field and safety routines.

Stage Two

A very similar approach for controlling Chip will be used in this phase of the competition Chip will move from place to place using potential field obstacle avoidance but the planner will choose goal locations in such a way as to systematically explore the entire competition area. Currently, dead reckoning is planned as the primary method for the robot to keep track of its location. Each object to be identified will be marked with a color coded sign that can be seen from all directions and color histogram visual identification will be used to find them As Chip moves about the competition area it will stop from time to time and look for objects. Each time a new object is identified the robot will speak its name and record its location in an internal map for use in stage three.

Stage Three

In this stage, Chip will be given the names of the objects it is to locate, look up their approximate locations in its map from stage two, and move to them using the same navigation routines as before. When the approximate location of each object is reached, Chip will search for the object visually and move up to touch it and say its name. The planner will attempt to find a reasonable route to take Chip from one object to the next.

The Team

The University of Chicago team consists of faculty members Jim Firby and Mike Swain. Also on the team are Mark Stricker, a postdoctoral fellow; Dave Christianson, an undergraduate student, and several graduate students who contribute to the effort occasionally.

CARMEL

The University of Michigan AI Lab

CARMEL is a Cybermotion K2A mobile platform with a ring of twenty-four sonar sensors. Motor control and sensor firings are controlled by two on-board computers. A third, 486-based, PC-compatible on-board computer runs all of the competition software and communicates with the other on-board computers. Object detection is accomplished using a color camera connected to a frame grabber that occupies a slot on the 486 PC. We have constructed a rotating table that will allow the camera to be panned without moving the robot; this will help reduce dead reckoning errors. All of these systems, plus the robot, are powered by two on-board twelve-volt batteries

Stage One

Over the last four years, researchers at the University of Michigan Mobile Robotics Lab have developed a reliable obstacle avoidance system (OAS) for fast mobile robots. This system has two major components: (a) a unique method for detecting and rejecting noise and crosstalk with ultrasonic sensors, called error eliminating rapid ultrasonic firing (EERUF); and (b) an obstacle avoidance method called the vector field histogram (VFH).

The innovative feature of EERUF is its ability to detect and reject ultrasonic noise caused by other mobile robots in the environment or by crosstalk (a phenomenon where one sensor receives the echo from another), caused by CARMEL's own sensors.

Since EERUF dramatically reduces the problem of crosstalk, CARMEL can fire its ultrasonic sensors at a rate of 160 ms per sensor, two to five times faster than in most other mobile robot applications

One of the most popular approaches to

obstacle avoidance is based on the principal of potential fields. However, while running this method at high speeds (0.6 – 0.8 m/s), instabilities occur with certain obstacle configurations. As a result, the vector field histogram (VFH) was developed. VFH uses the ultrasonic range data to continuously update a grid-type world model, called the histogram grid From the histogram grid, the VFH method computes a polar histogram that holds information about the obstacle distribution around the robot

The combination of EERUF and VFH is uniquely suited to high-speed obstacle avoidance; it has been demonstrated to perform reliable obstacle avoidance in the most difficult obstacle courses at speeds of up to one m/s (3.3 feet per second)

In addition to VFH we are using a global path planner that searches the certainty grid and creates a list of via points (intermediary goal points) that represents the shortest path to the goal.

After the via points are determined, VFH is directed to go to each of the via points in turn This combines global path planning with local obstacle avoidance and helps to avoid traps and dead-end situations. At the highest level, CARMEL will be directed to follow a "lawnmower-like" search pattern that will sweep the entire ring during the course of stage one. CARMEL will continually change the search pattern until the twenty minutes allotted for stage one are completed.

Stage Two

The basic obstacle avoidance and path planning algorithms used in stage one will be augmented by visual cue detection and absolute positioning algorithms for stage two.

We will tag the objects with horizontally striped tubes, slightly larger than three inches in diameter so as to fit over the PVC pipes used in the competition. The striped tubes have six orange stripes which delineate five white stripes. Each of the white stripes represents a bit. Filling in a white stripe with orange (thus making an orange stripe that is three times larger than normal) means that that bit is "on." We have ten tubes with ten different bit patterns for the ten objects.

Our algorithm for detecting objects is broken into two parts. First, the image is filtered for red, which extracts the orange stripes very nicely. Then a one-pass algorithm goes down each column of the filtered image, looking for the characteristic striped pattern. Adjacent patterns are combined and the size of the pattern is used to estimate the distance to the object Preliminary results show the algorithm to be fast and reliable at up to eight meters (twenty-six feet).

During the course of twenty minutes of intensive navigation, we expect CARMEL's dead reckoning sensors to develop inaccuracies Luckily, we have ten natural landmarks, the objects, which we can use to triangulate CARMEL's exact position and orientation and update its dead reckoning sensors. First, we need to pin down some object locations precisely at the beginning of stage two, when we know the robot's position accurately. We plan to take multiple sightings of objects from different positions in the ring and triangulate from those sightings to get exact object positions.

Next, we have developed an algorithm that will take any three object headings with respect to the robot and use those objects' known locations in the ring to determine the robot's location and orientation. Several methods to do three object triangulation are known in the literature, including iterative search, geometric circle intersection, geometric triangulation, and Newton-Raphson. In tests we conducted, all of these methods were found to fail with certain combinations of object locations. However, we were able to combine each method to overcome their individual weaknesses. Preliminary tests show that our implementation is robust with respect to errors in object location.

CARMEL will follow the same basic exploration pattern in stage two that it did in stage one. It will go up and down in the ring in a "lawnmower" fashion, stopping periodically to do a visual scan for objects. Upon finding an object, CARMEL will attempt to approach within two robot diameters (RDs) of the object and announce that it has found an object. This object is then given an *(x,y)* location within the ring, as determined by its distance and orientation from the robot. This *(x,y)* position is updated with multiple sightings of the same object. In addition, CARMEL will note those locations where three or more objects are visible, as it can later return there to update its dead reckoning sensors. CARMEL will continue this systematic search until it has found all ten objects or until it needs to retreat and update its absolute position (we are experimentally determining how far CARMEL can travel before its dead reckoning sensors become too inaccurate for efficient navigation). After finding all ten objects, CARMEL may return to some of them to more precisely fix their location in the ring in anticipation of stage three.

Stage Three

If stage two has gone well, stage three will be simply a matter of computing a path to each of the three objects in turn, moving to each of them, taking an image at each to verify that the object is indeed there and then returning to the start location. If CARMEL did not get a chance to fix all of the object locations precisely in stage two, it may need to perform some more extensive visual searching in stage three. The certainty grid will be saved from the end of stage two, allowing the global path planner to compute a shortest path to each goal location. Of course, VFH will be running, in case an obstacle has been jostled out of position.

The Team

Our team is almost entirely composed of students and run by students. There are thirteen graduate students and five undergraduate students. In addition, two faculty members of the Artificial Intelligence Lab serve as advisors and three research associates provide additional technical assistance.

Demonstrations

ATTILA

MIT Artificial Intelligence Lab
Massachusetts Institute of Technology

Attila is a small six legged autonomous robot. It was designed and built under the supervision of Prof. Rodney Brooks in the Mobile Robot Group at MIT. The robot measures fourteen inches long, stands six inches high and weighs six pounds. It has six 3 DOF legs and one 2 DOF antenna. Despite its small size, Attila currently has twenty-four actuators and over one hundred sensors of fourteen different types all connected via a local network to eight on board computers.

Attila's Sensors

Attila's sensors are organized into two groups: low level sensors and high level sensors. The low level sensors are the most reliable and accurate sensors on the robot. They are specific in what they sense, and the output does not need much interpretation. The high level sensors are more complex, require more interpretation of the outputs, and are less reliable than the low level sensors. The organizational principle of the sensors was for the robot to be fully operational with the reliable low level sensors and to use the high level sensors to improve performance. Here's a summary of the sensors on Attila.

The sensors for the robot were chosen to give the robot a multi-range view of its environment. The long range IR range sensor and CCD camera could give the robot a sense of what the upcoming terrain looks like and give it a chance to steer itself towards a promising route. The antenna helps the robot steer around local obstacles The low level sensors helps the robot choose foot placements. Together these sensors act as a terrain filter that serves to guide the robot through plausible routes.

Subsumption Architecture and Behavior Language

The MIT Mobile Robotics group introduced Subsumption architecture in 1985 as a radically different control architecture for intelligent autonomous agents. It also introduced Behavior Language in 1990 as a high level programming language for implementing Subsumption architecture. The Subsumption architecture divides the architecture into task-achieveing modules (also called behaviors). As a result, the problem is sliced vertically into parallel task-achieveing modules.

Each slice in the vertical division is layer of competence. The main idea is that we can build layers of a control system corresponding to each level of competence and simply add a new layer to an existing set to move to the next higher level of overall competence The lower layers run continually and are unaware of higher layers. However, when the higher layers wish to take control they can subsume the roles of lower levels by supressing lower level outputs.

At the lowest level, layers are built of behaviors which run in parallel, perform their own perception, and send messages to each other over virtual wires. Each behavior can be viewed as a finite state machine with input wires, output wires, and the ability to hold some data structures. The behaviors run completely asychronously, monitor their input wires, perform computation, control actuators, or send messages out their output wires. Individual behaviors are connected to form task-achieving modules, and these task-achieving modules can be grouped together to form layers.

Behavior selection is a central issue when designing behavior based systems. Behavior based systems are made up of several task-achieving behaviors. The system is designed to activate only behaviors that are relevant to the robot's situation at any point in time. Typically, the system uses sensor values to select and activate behaviors. It is also possible for behaviors to activate other behaviors directly by sending messages.

Behavior conflict issues arise when designing behavior based systems. It is important to design the system such that conflicting behaviors are not active at the same time. To deal with this situation, behaviors have the ability to inhibit outputs or suppress inputs of other behaviors. Inhibition of outputs is used when the inhibiting behavior does not want the inhibited behaviors outputs influencing the system. Behaviors can prevent other behaviors from becoming active by suppressing inputs used for activation of other behaviors.

In Behavior language, behaviors have an activation energy level and a threshold. When the activation energy level of a behavior is above the threshold, the behavior is active. If behaviors are defined as haltable processes, the behavior does not run unless its activation level is above the threshold. If behaviors are defined as inhibited processes, the behavior always runs, but the outputs are inhibited unless the activation energy is above the threshold. Behaviors can send or remove activation energy from other behaviors. On a larger scale, the behavior activation mechanism is useful for activating behavior networks.

Attila's Control Structure

A control network is currently composed of three types of behaviors: calibration behaviors, virtual sensor behaviors, and motion behaviors.

As time passes, the robot's analog sensors will drift. The function of the calibration behaviors is to periodically update the reference value, maximum value and minimum value of various analog sensors (position sensors, force sensors, and velocity sensors). To accomplish this, the robot pauses and activates its calibration behaviors at scheduled time intervals. The calibration routine consists of the robot exercising its sensors and recording the appropriate values.

The virtual sensor behaviors are responsible for processing sensory data. Inputs to these behaviors are actual sensor data and calibration values, and outputs from these behaviors are virtual sensor data. For example, signals from force sensors, velocity sensors, contact sensors, position sensors, etc. are types of input The virtual sensor behaviors combines and processes the actual sensor data (and perhaps compares the result to calibrated values) to produce virtual sensor data such as "leg collision," "foot has ground contact," "hole in terrain," etc . The virtual sensor results serve as inpus to the motion behaviors. The virtual sensor behaviors are also responsible for behavior activation. If a virtual sensor is "true," it activates the appropriate behavior or groups of behaviors. For example, if the "step in hole" virtual sensor of a leg is "true" the virtual sensor behavior will activate the "squat" behaviors on the other legs to lower the body.

The motion behaviors are responsible for actuating the robot based on virtual sensor inpu and taking care of behavior conflicts by deactivating other behaviors that move the same actuator. At a low level such as walking, the behavior deactivation relationships are hardwired into the network so that certain behaviors always take precedence over other behaviors. With higher level behaviors, a more flexible network may be desirable.

The three types of behaviors are connected to form a control network. Identical control networks exist on each of the legs of the robot. The body has a different control network. Behaviors on a leg can activate, deactivate, send outputs, and receive inputs from behaviors on other legs. Behaviors on the body can also activate, deactivate, send outputs, and receive inputs from the legs. From the interaction between the control network on the legs and body, various walking behaviors emerge.

The Team

This work was done by Cynthia Ferrell under the supervision of Prof. Rodney A. Brooks in the Mobile Robotics Group at MIT

BERT AND ERNIE

AP Group , MIT Artificial Intelligence Laboratory

Bert and Ernie are sensor robots designed by Fred Martin and Randy Sargent of MIT's Media Lab. The robots are seven inches long and five inches wide. The height varies from four inches in the front to five inches in the back.

The robots are propelled by the two rear wheels which are connected to the motors. There is a caster on the front of the robot. The robots were designed with a number of sensors, including four touch (bump) sensors (left front, right front, left rear and right rear); four bend sensors (used like whiskers, mounted on sides of robot near each corner); three photocells (right front corner, left front corner—both bent forward), and one in the rear facing up for room lighting detection; one inclination (tilt) sensor; two shaft encoders; four IR sensors (Sharp GP1U52Y) and eight IR LEDs - sensors are placed in the middle front, middle back and at a forty-five degree angle towards the front on each side; one IR feedback sensor; a battery level; a pyro electric sensor (not installed); and two floor reflectance sensors (not installed). There is also a microphone and speaker for communication with the outside world.

The robots have the ability to communicate with each other through a radio transmitter and receiver. These boards only have a four bit capability. There is also a transmitting station to communicate with the robots.

Usually, robots are provided with a language through software written by humans. By allowing the robots to create their own meanings for signals, the robots may develop a language that is far better suited to their tasks than a language conceived by humans.

Currently, the task to be achieved is coordinated movement. The robot hardware itself is homogeneous, but we have made them heterogeneous by only telling the task to one of the robots. The robot receiving the task information (Bert) needs to select actions from its list of possibilities. One set of actions contains the possibilities for ways to move; the other set deals with sending signals to the other robot (Ernie). Ernie will have a set of actions that it will choose from upon hearing a signal.

At first, both robots will choose actions randomly. We will use reinforcement learning to help the robots converge on a language. The robots will be reinforced by a human; the understanding of reinforcement signals will be hard coded into the robots. Later, we hope to extend this work by having the robots reinforce one another. Initially, the robots will develop signals for "move right" "move left" and "move straight." Once the initial work has been done, we want the robots to develop a compositional language. In a compositional language, each word has a meaning and has a way to relate to other words. The first language developed by the robots will be non-compositional—the robots are learning "move right," "move left," and "move straight." In the compositional language, the robots will have a word for "move" and the qualifiers "right," "left," and "straight." In the demo portion of the exhibition, we will show the current state of communication between Bert and Ernie. Although all communication takes the forms of radio signals, we will use the speakers also so people can hear representations of the different signals being sent.

The Team

The demo is part of Holly Yanco's graduate research under Prof. Lynn Stein at MIT's Artificial Intelligence Laboratory in the AP Group. Much of the hardware work has been done by Tim Tang, an MIT undergraduate.

The Autonomous Aerial Robot

Georgia Tech Unmanned Aerial Vehicle

The system is built around a remotely contolled helicopter offered as a kit from Minature Aircraft Company. The helicopter, an X-Cell 60, weighs ten pounds empty and can carry up to twenty pounds while hovering. The maximum forward speed is about 100 MPH. The rotor diameter is almost six feet, and overall length is six feet, five inches. The onboard stability and control system is built using a 68332 business card computer and two six-axis sensor units as well as altitude sensors. Roll, pitch, and yaw rate as well as pitch, roll, and yaw position is measured and controlled via an inner loop, and movement around the competition arena is controlled with an outer loop that is cued from a groundbased mission planning computer that integrates position information. The navigation system consists of three vision systems that track a pattern on the helicopter and triangulate the x and y position to feed into the mission planning software. Two of these integrated vision systems are mounted on servo motors to expand the field of view availible to the cameras. The retrieval system consists of a motor to lower a mechanism into the ground bin, and two interchangable retrieval mechanism concepts. One is a magnetic array that is able to sense a disk, and then energize a particular electro-magnet. The other is a tethered sub-vehicle based on a model car with a search pattern and a method a grasping the disk.