AIAA-94-1252-CP

SIMPLIFYING APPLICATIONS SOFTWARE FOR VISION GUIDED ROBOT IMPLEMENTATION

Charlie Duncheon
Adept Technology, Inc.
San Jose, California

Abstract

This paper begins with the background on how robotic implementation has flourished in Japan while only moderately growing in United States manufacturing. Contributing reasons are provided with focus on the constraint of the time and difficulty of robot applications software. Particular focus is made on the largest robot application segments in the world markets, material handling and assembly. Manufacturing demands of the 1990's are cited and scenarios of 21st century plants are described. Successful implementation of vision guided robots for these plants is described by use of the following system building blocks:

1. An industry standard form factor, open architecture hardware control platform.
2. A proven and integrated real time operating system and factory automation language within the platform that provides direct support for sophisticated motion control and real time sensing.
3. A building block, icon/menu based application software approach for both developing applications and for easy factory floor interface.
4. An open architecture platform that allows other proven operating systems and/or cell control hardware/software solutions to coexist on the same back plane with the core motion/vision operating system.

Background

Despite robot technology being invented in the United States, the implementation rate in the US market fell behind that of the Japanese by tens of thousands of robots per year by the end of the 80's. The trend continued in 1993 despite a new parity in the cost of capital between US and Japan. New "Agile Manufacturing" thrusts have emerged in US manufacturing yet they are noticeably short of true flexible automation strategies. In the early 80's Prudential Bache projected a US robot business of $2 billion by 1990. The market actually reached $500 million in the early 90's. The current installed base of industrial robots in Japan is 400,000 units, while in the US it is only 50,000 units. Japan installs more robots per year than the total installed base in the US.[1]

Why the slow adaptation of industrial robots and flexible automation in the US compared to the continuing robust growth in Japan? First of all, Prudential Bache was correct in identifying $2 billion of robot applications that should have been implemented in the US. In fact, it would have been even more had the technology been implemented at the Japanese rate. The fundamental constraint to growth of the robot industry was the difficulty in implementing the technology. Suppliers overestimated the time, capacity and technical skill level manufacturers had to implement automation. Financial justification followed the same parameters as applied to traditional hard automation, leading to only the most challenging applications gaining financial approval. This only compounded the problem of successfully implementing the technology in manufacturing.

A study by Adept Technology, Inc.[2] showed that in the 1980's the cost to design, write, debug and document application software was as high as 50% of the total robot system costs. And for the most part the software developed was custom to each application. Application software development, debug, and implementation was also determined to be a common bottleneck to system implementation schedules. Other robot companies and robot systems integrators have come to the same conclusions and have taken steps to standardize application software. In 1992 and first half 1993 statistics released by the Robotics

Industries Association show healthy increases in the implementation of robots in the US market, due in part to more enabling software. 1992 orders for US based robot manufacturers grew 21.5%, and first half 1993 orders jumped 40%.[1]

This paper will discuss how the utilization of **existing** and **proven** hardware and software modules which minimize custom software development cannot only increase US robot implementation beyond the current improved rates, but put the US in the global lead. There will be no formulas or equations; manufacturing engineers who are our vital hope for competitiveness have no time for such. They have plenty of their own process algorithms to address. They care about practical suggestions that allow them to address the cost and quality issues without compromising lead time to market. That is what this paper will attempt to address.

Manufacturing Forces of the 90's

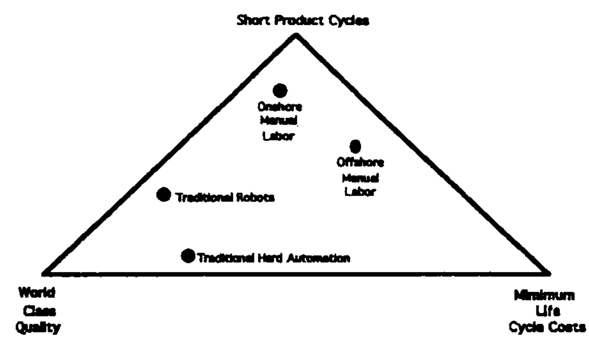
There are many forces influencing manufacturers today but three main forces stand out that will determine global manufacturing strategies:

1. World Class Quality - World class quality is not an option but the price of admission for those manufacturers who want to survive the decade. The "Six Sigma" commitment by Motorola is one of the more publicized commitments to this imperative.

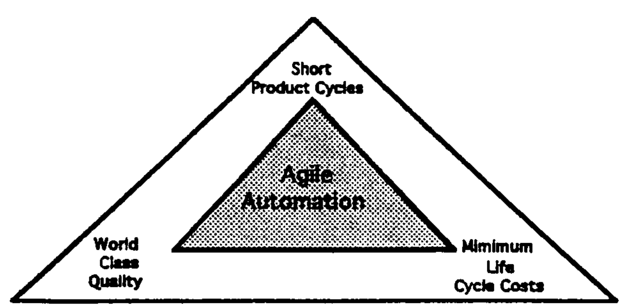
2. Minimum Life Cycle Costs - The 90's is being dubbed as the "value decade", as manufacturers constantly evaluate their manufacturing costs against global competitors. As governments realign the world markets via NAFTA, GATT and EC agreements, even those manufacturers who limit themselves to domestic markets will see global competition in their home markets.

3. Minimum Product to Market Times - The laptop computer used to write this paper is currently nine months old and has already been obsoleted by a higher performing model. Conventional high volume, low mix production strategies are in conflict with this force. Manufacturers are being challenged to have shorter changeover times or running multiple products simultaneously on the same line. One power supply manufacturer has an objective that a customer can specify a custom power supply with a lot size of one and receive it within one day, built with flexible automation.

Unfortunately, the above three forces have traditionally been in conflict with each other. Automation might have addressed quality forces but did not support rapid lead time to markets. Traditional robot automation may not have met cost objectives. And many US companies found that offshore labor did not meet quality requirements.



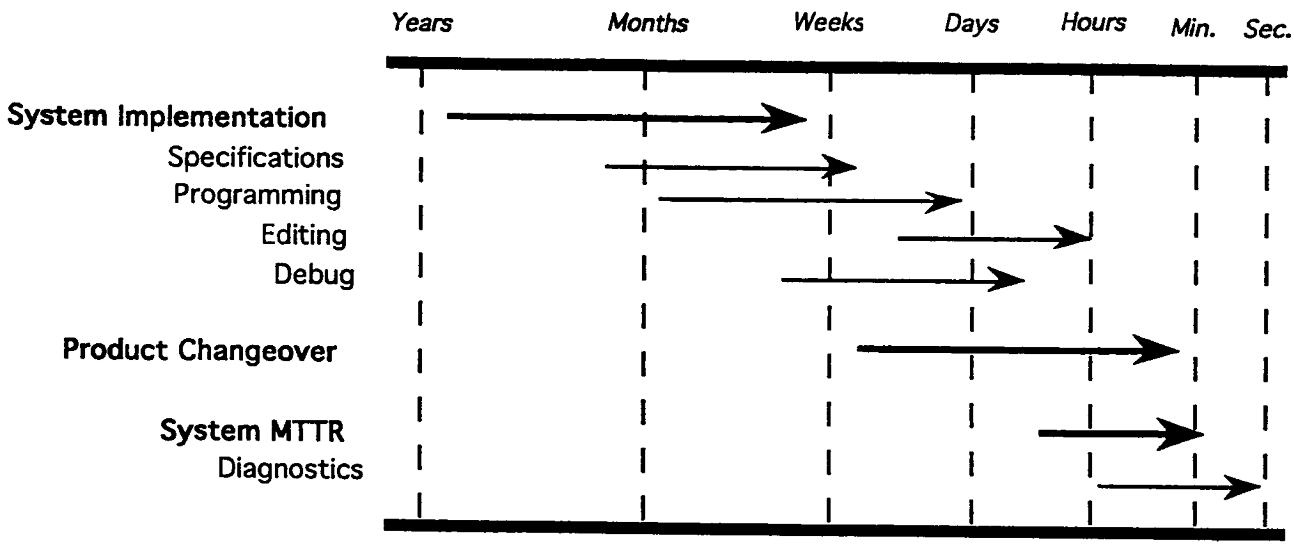
The RIA and its member companies believe that there does not have to be a tradeoff among these forces. The belief is that properly applied, "Agile Automation" can "shrink' the triangle of tradeoffs.



Adept Technology, Inc has developed a long range strategy for minimizing flexible automation cell implementation and changeover time with the development of vision guided flexible feeding technologies and modular functional and application software. These

Rapid Deployment Automation

Objective: Conservation of Time



efforts are part of Adept's overall vision of "Rapid Deployment Automation".

Rapid Deployment Automation is an overall strategy focused at reducing the time, and thereby the cost, required to implement flexible automation. Given that systems typically cost 2x the direct hardware costs, Adept feels strongly that reducing engineering content will be a primary driver to reducing the cost of flexible automation. Rapid Deployment Automation also address directly the product changeover / time to market issues discussed above.

Flexible feeding is a key element in Rapid Deployment Automation, as by nature it reduces the amount of hard tooling in a robot cell, which reduces costs and promotes rapid development of cells.[3] Flexible feeding replaces traditional part specific feeding systems such as bowl feeders and precision dunnage with intelligent sensor based robotic application software. Sensing is provided by integrated machine vision and real time force sensing, which are used to locate random or loosely oriented parts on simple conveying or infeed systems. Flexible Feeding software includes modules and algorithm's to locate the parts, determine their orientation, control the conveyors and recirculating devices, and acquire and place the parts. The Flexible Feeding concept, and the specific software modules that drive it, are examples of how software can dramatically simplify the overall engineering task in a flexible automation cell.

21st Century Factories

If Adept and other RIA companies are successful, the following scenario is possible for the year 2000 manufacturing assembly plants:

1. Small, nimble plants, less than 250 employees, located close to consumer bases of population or major OEM customers.

2. Lot size of one, same day delivery to customers. Rapid changeover from one part to another and rapid implementation of incremental process and component part improvements and changes.

3. Vision guided robot and flexible feeder based production. Limited investment in hard tooling. Intelligent software based feeding and fixturing.

4. No software language based programming within factories. Simplified set up of highly intelligent software modules with imbedded process knowledge by floor personnel.

How Do We Meet the Challenge?

One way we will not meet the challenge is to continue to develop application software and/or motion control platforms "from scratch" because of what appears to be unique or process specific requirements. The development cycles for applications will exceed the product cycles themselves. The approach certain robot and systems suppliers are taking is to provide function and application software modules that serve as building blocks to the final line application program.

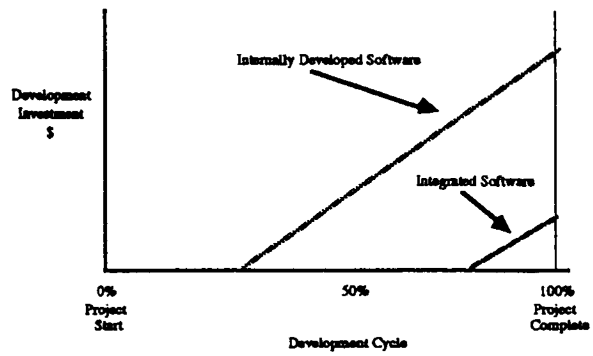
This modular approach to software development is not new. There are countless libraries today of software functions from various hardware and software vendors. The burden in using modular libraries is in the linkage between modules. This fine tuning still requires highly skilled programmers, and significant time. The new approach is to provide these modules within a structure or framework that defines how the modules work together. Unlike object oriented programming, which offers a similar strategy, this new generation of factory automation software includes high level interfaces and process knowledge about the tasks themselves to allow setup by non programmers.[4]

The best way to describe such an approach is with an application. Assume we have a dental adhesive applicator product with a cylindrical body with one open end, an impeller that must insert into the body, and a cap that must be inserted at the top of the body and impeller. The body arrives open end up in multiples on a tray moving on a roller conveyor and stops against an activated fixture. The dunnage that carries the body is general purpose, and does not provide significant precision in locating the bodies. The manufacturer is dealing with a product with a short life cycle and wants to quickly reuse the robot, grippers, feeders, and software when any of the three parts change in design. A single vision guided flexible feeder for the impellers and caps, and vision guided registration of the bodies in the trays along with inspection for acceptable inner body diameters are among the cell design parameters.

The long path to a solution would be to interface one supplier's vision system and language with a second motion system and language, with custom software written in C, perhaps on a PC platform. Not only is the development cycle long, but history has shown debug activity for this approach to extend well past floor implementation.

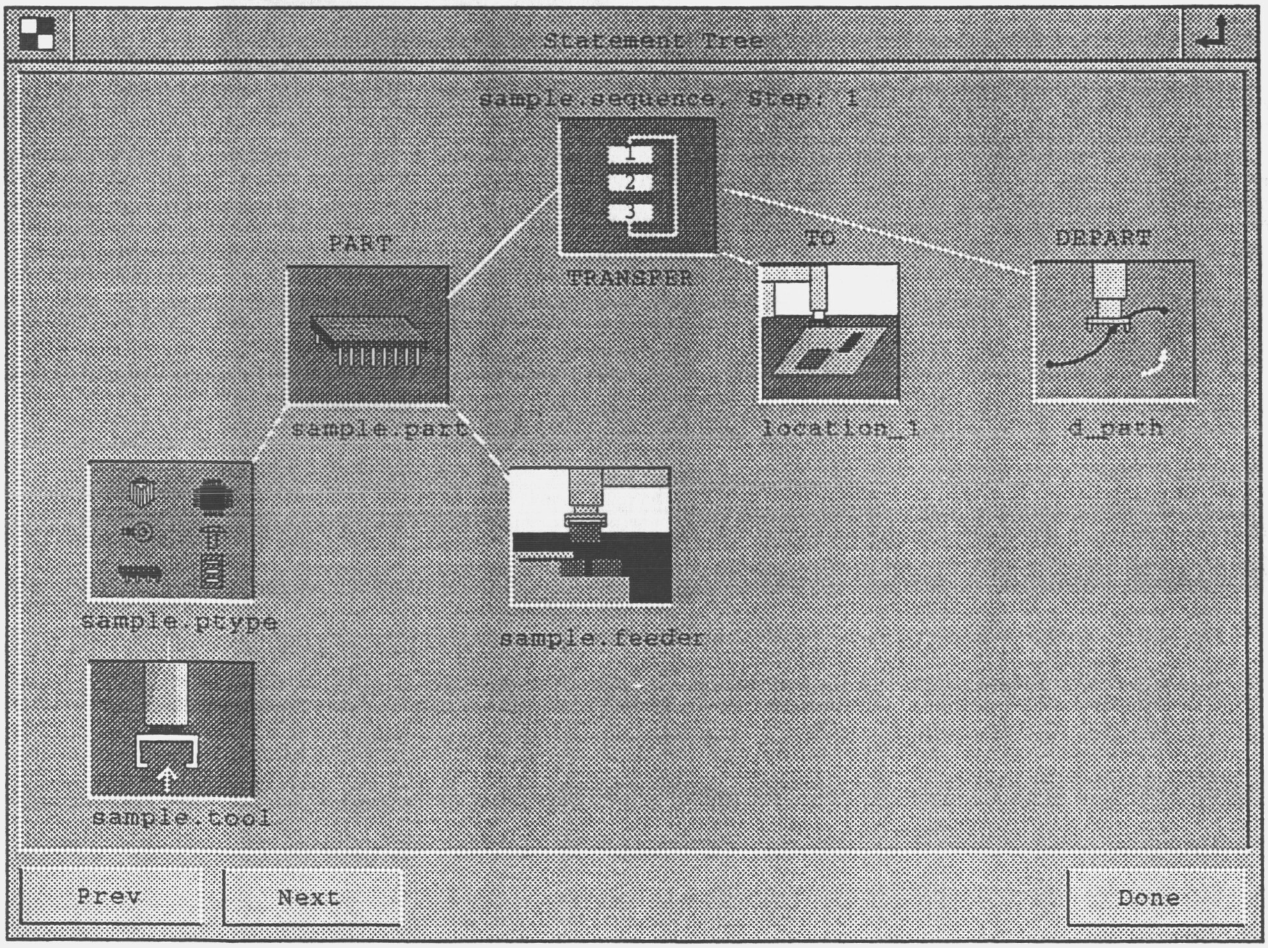
This potentially ill fated path relates to a lack of empathy of many developers with regard to "real time" in a robot cell on the factory floor vs. that in the work station environment. Deterministic, multitasking operating systems that can do context switching in micro seconds are crucial when expensive grippers with expensive parts are on a collision path. More important than part and gripper damage, system downtime is unaffordable. Manufacturers like Toyota and Michelin are requiring 40,000 hour MTBF performance from equipment suppliers. In other words, bugs or errors at the work station are manageable, but unacceptable on the assembly floor.

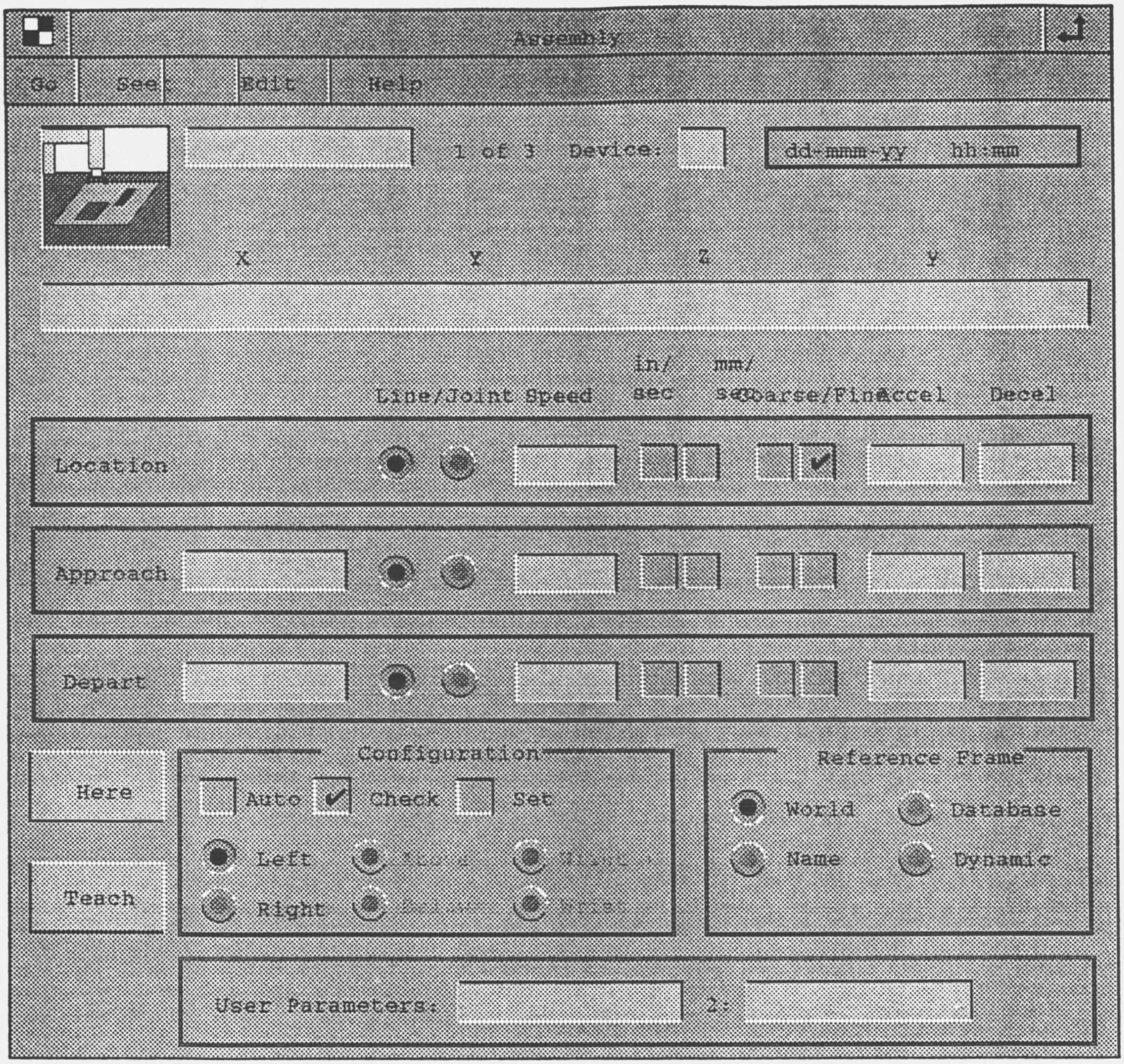
The integrated control platform offers a much lower cost, schedule time, and risk approach to this solution. A platform that has thousands successfully running applications while keeping motion, vision and force sensing, and communications software under one language is the optimum place from which to start for this application.[5]



An Adept control platform that control's Adept robots or any other robot mechanism would be made up of "wrung out" functional software modules that build into an overall applications module. These functional modules are menu driven and utilize a common data base structure and common global variables. Modules are tied together to create the specific application module, imbedding any necessary process knowledge, by use of the high level V+ programming language. Adept's standard functional modules and tools for vision and force guided assembly are represented by icons.

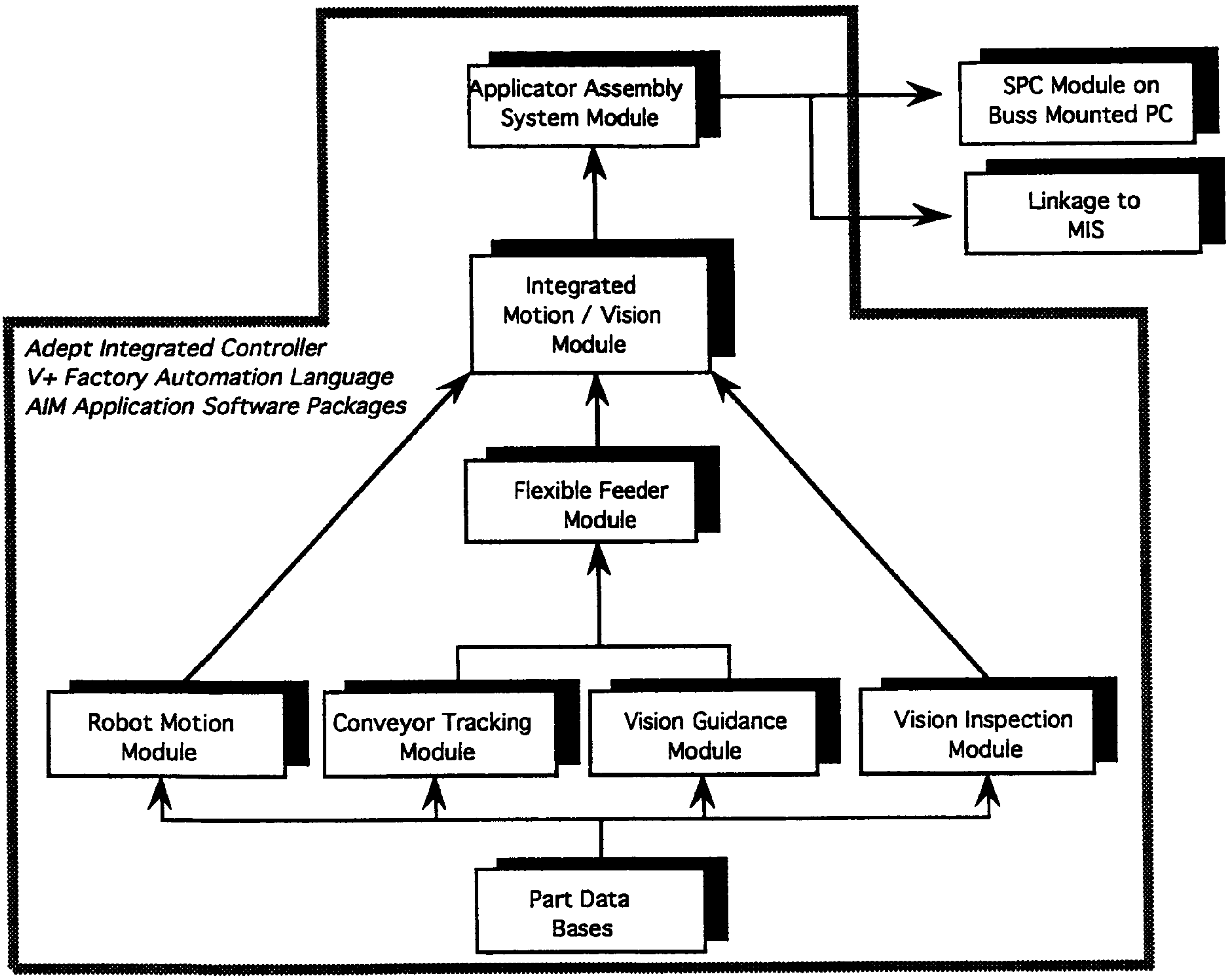
Adept's patented AIM [tm] (Assembly Information Manager) environment allows the developer to create additional linked icons as required to represent combinations of Adept modules and any application customization or process imbedding.



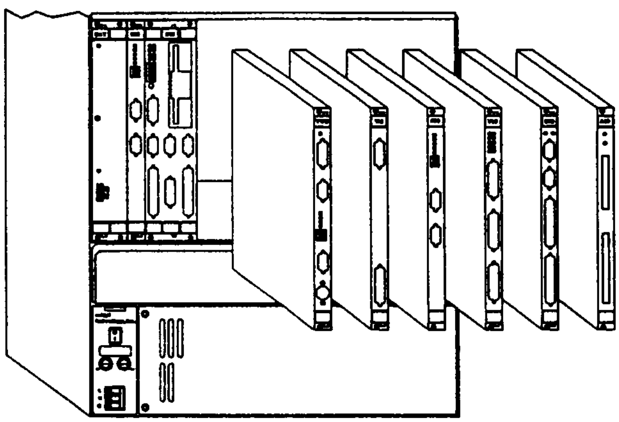


Assembly
Go
Seek
Edit
Help
1 of 3
Device:
dd-mmm-yy hh:mm
X
Y
Z
Y
in/ sec
mm/ sec
Line/Joint
Speed
Coarse/Fine
Accel
Decel
Location
Approach
Depart
Here
Teach
Configuration
Auto
Check
Set
Left
Right
Reference Frame
World
Database
Name
Dynamic
User Parameters:
2:

Example of Robot Cell Software Structure



So the impeller assembly application can be quickly developed with bug free software modules which also leave an rational user interface for modifying on the factory floor. Now let's assume the manufacturer has strict FDA requirements for Statistical Process Control records and has invested heavily in a PC based SPC program that he wants to apply to any workstation. The Adept MV controller contains a standard VME back plane with open slots for other boards. The manufacturer can have a VME board with an imbedded PC inserted on the Adept back plane and communicate as required with the Adept operating system and AIM. Customization is limited to defining what variables are needed by the SPC module.



Summary

The approach described here is simple: Use commercially available software and hardware wherever possible to minimize system costs, schedules, and risks. There is an abundance of software and hardware technology that does not have to be reinvented. This approach is what put the Japanese in a leadership role in the 80's. The US now can do the same with a much more flexible array of vision guided robot technologies.

1. Robotics Industries Association, Ann Arbor, Michigan.

2. Adept Technology internal integration study. November, 1989.

3. G. Orelind, "Flexible Part Feeding for Robotic Workcells", Robotics World, Summer 1993.

4. J. Campbell, P. Cencik, "General Purpose Scalable Motion / Vision Controllers", Proceedings, ISTA Conference on Lean Manufacturing, August 1993.

5. J. Campbell, "The Economics of Utilizing an Integrated Control Platform for OEM Motion Control", Motion Control, February 1993.