The ADACS Implementation of the UTAP Architecture

Robert T. Russell, Jr., John Michaloski and Keith Stouffer
Intelligent Systems Division
National Institute of Standards and Technology

Summary The Advanced Deburring and Chamfer System is robotic workcell designed to put precision chamfers on parts made with hard metals. The project was funded by the United States Navy so that the technology developed could be applied to their air defense systems. The Unified Telerobotic Architecture Project was a United States Air Force funded project to develop an open architecture for robotic applications for servicing aircraft. The main focus was to develop an architecture with standard interfaces so commercially available components could easily be substituted. This paper will describe how the ADACS workcell controller was implemented through the use of the Unified Telerobotic Architecture Project's architecture and interfaces.

1 NOTATION AND UNITS

ADACS refers to the Advanced Deburring and Chamfering System. UTAP refers to the Unified Telerobotic Architecture Project. RCS refers to the Real-Time Control System. CADET refers to the Chamfering and Deburring End-of-arm Tool. NCFS refers to Numerically Controlled Finishing System.

2 INTRODUCTION

The finishing operation is a critical step in the manufacture of parts made with hard metals. Many thousands of dollars are spent on machining processes for precision applications only to be handed off to a human at a burr bench to perform a finishing process. This hand-crafting tends to be inaccurate and produces a high scrap rate which in turn means that the money spent on machining is also lost.

ADACS augments the human with a numerically-controlled finishing system designed for putting precision chamfers on aircraft engine components and helicopter parts. These aircraft components are made of hard metals such as inconel and titanium that require special tooling and exacting procedures to produce suitable chamfers. The tolerances on many of these parts approach 0.07mm.

As a numerically-controlled finishing system (NCFS), ADACS is a study in integration of computer and control subsystems. ADACS has CAD modelling; graphical simulation, real-time motion control; adaptive, force-based tooling; and operator-supervisory control. Current proprietary or closed controller technology does not span the breadth of the application and also precludes the tight coordination of force-controlled tooling with motion control required in chamfering. The primary upshot of these constraints was the need for ADACS to use and integrate off-the-shelf-components from differing vendors in a distributed environment. The integration of such differing system components requires a sound design approach or implementation of a point-solution can result.

An open system built from open component technology was the major design paradigm used to achieve the ADACS system requirements for integration, flexibility, and extensibility. System design using an open architecture reference model with well-defined interfaces offers a sound approach to implementation that can be adapted to satisfy future requirements. ADACS used the Unified Telerobotic Architecture Project (UTAP) as the reference model in which to construct the NCFS. UTAP is an open-architecture reference model with well-defined interfaces. This paper will show how the ADACS was designed and implemented used the UTAP architecture and the UTAP component interfaces as a guideline.

The paper is organized as follows. The next section will present an overview of the chamfering application. For chamfering applications under numerical control, the integration of component processes will lead into the discussion of the UTAP reference architecture as it applies to the ADACS application. Next, a discussion of the system design will be presented which will include the hardware, software and component integration. Finally, the conclusion will discuss the merits and potential for the UTAP design approach.

3 APPLICATION

3.1 Automating the Chamfering Process

Parts manufacturing has evolved to the point that most parts are designed using a CAD system. To machine the part, CAM machining process parameters (e.g., specific feeds and tooling) are combined with a CAD model to generate a part program tuned to run on an actual machining workstation. This marriage of CAD and CAM is sometimes done automatically by computer-generation or may occur with the help of engineering-supervised computer-generation.

Part finishing could greatly benefit from supplementing the process with engineering-supervised computer-generation to combine CAD data with finishing process parameters. Unlike machining, however, chamfering uses different heuristics and domain expertise in performing the chamfering process. Chamfering requires a skill level that currently cannot be reasonably duplicated by complete computer automation. Instead, the ADACS finishing process makes use of the CAD data that will utilize the expertise of engineers and chamfering operators to tune the finishing process to a specific part. Figure 1 diagrams the NCFS data flow within the computer-assisted finishing domain as implemented in the ADACS.

Data is defined, translated, and merged in several steps along the NCFS process. Part data from the CAD software goes to a post processor to extract the feature information and process information. This information is used by the next level down to produce a motion plan and tool parameters that correspond to the motions in the plan. From here the segment planner breaks the motion up into smaller segments to send to the robot and the tool parameters are set and maintained through closed loop control. This is explained in greater detail in Stouffer et. al. (1).

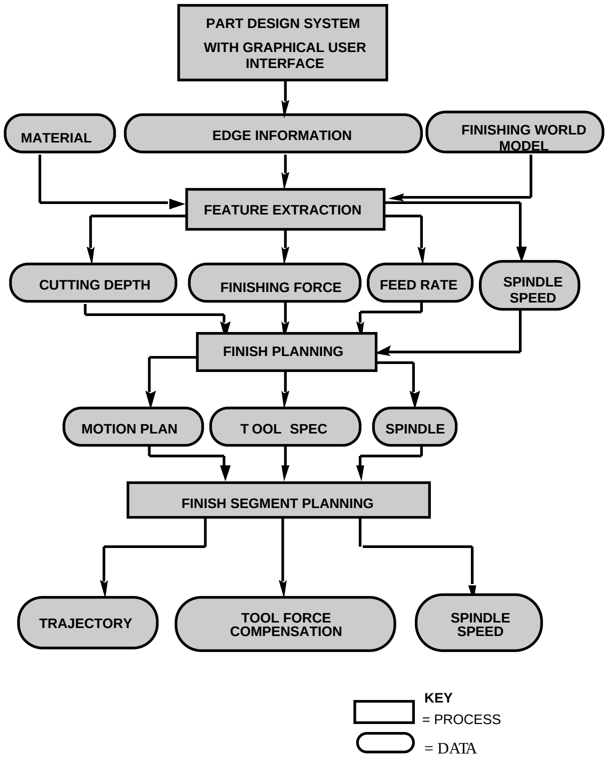

Figure 1: ADACS Finishing Process Model

3.2 Typical Scenario

The ADACS is designed to have two operator interfaces. One is for the manufacturing engineer who understands the process and can assign chamfer depths, speeds and to specific part edges where the chamfers should occur. He produces a part program for the shop floor to use. This is done on a CAD package in which he can select edges and give parameters to use on the edge. Next he runs the part information through a workcell simulation to verify that no unexpected collisions will occur between the tool and the part. The part program is used by shop floor personnel operating the actual workstation through the second operator interface. This interface allows the operator to select a part program, run it and change various settings (feed rate, force, etc.) on the fly. The part program is interpreted by the controller with the motion control aspect of the NCFS behaving as outlined in the previous section.

4 SYSTEM DESIGN

4.1 Architecture

Benefits of open architectures are numerous and include lower-cost, flexibility, extensibility, interoperability, and portability. The UTAP architecture goal is to accommodate different types of kinematic devices - both machining and material handling - with different degrees of freedom, accommodate different part materials and part geometries, new tasks in the workplace, and provide a facility to upgrade/change equipment, sensors, and feedback mechanisms as technology advances. The UTAP architecture is described in more detail in NASA (2).

The UTAP reference model architecture serves as a guide as to how to structure the components in a system - not as a mandate. UTAP assumes a similar architecture for all modes of control: teleoperation, shared control, supervised autonomy and autonomous control. Depending on the application, a similar, but not necessarily duplicate instance of the reference architecture may be developed. The goal of the reference model architecture is to provide a general model for defining the relationships among elemental components in a system.

The UTAP architecture as shown in Figure 2 describes a connected set of modules consisting of hardware components and a software architecture built from software components. The hardware components are distinguished as physical hardware items that might be purchased. The software architecture is separated into functional types of software modules, the software modules themselves, and the application programs.

Application programs execute in a distributed fashion with a parent module controlling a set of subsystems that are built from a set of component modules. An application program is separated into a parent task program and into subsystem task programs. A subsystem is characterized by having a separate task program. There may be separate task programs running on the same or separate platform or control devices. Coordination between separate task programs is achieved by direct communication between the subsystem task programs and/or through communication with a parent task program which communicates with the subsystem task programs to coordinate their control.

Within the UTAP architecture, software modules have a domain of functional capability. The ADACS functionality for CAD and simulation is composed as several modules in the UTAP architecture including - Task description and supervision, Object Knowledgebase, Object Modeling, Task Knowledgebase, and Subsystem Simulation.

The bridge between the engineering and process description is done by the Task Program Sequencing module which is responsible for parsing the part program and interpreting its contents. From this information it produces commands to the Task Level Control. The Task Level Control is responsible for coordinating control of the robot(s), and tool(s). It also includes the trajectory generation functionality. It reads data from sensors and maintains a database of this knowledge. The robot and tool servo modules are responsible for robot configuration commands, tool control commands, and status.

ADACS uses feature based knowledge of the part. Each feature has associated robot motions and tool parameters. It is critical that the correct tool parameters are executed for the corresponding robot trajectory. Otherwise the tool may be moving in free space chamfering air or worse yet, damage may occur to the part. As goal points are placed on the trajectory generator queue, a corresponding queue in the tool controller is receiving tool commands that contain a marker for the robot motion for which each set of parameters is to be executed. The tool controller continuously monitors the robot motions and sets the parameters when necessary.

4.2 Hardware Design

The approach used in the ADACS is called “around the arm control.” It encompasses the use of one manipulator for gross positioning and one for fine positioning. This is described in Murphy et. al. (3). The ADACS is a robotic workcell consisting of a six degree of freedom articulated robotic arm and an active force controlled chamfering and deburring tool. This setup allows the robot to position the tool tip close to the part and the tool can then maintain contact as the robot travels through its trajectories. The ADACS control structure is shown in figure 3.

The CADET is an active chamfering and deburring tool that uses force feedback to achieve precision chamfers. It was designed by United Technologies Research Center. The design is such that the tool is balanced so that no matter how it is oriented there is no need for gravity compensation in the force feedback loop. It also uses voice coil actuators so that is inherently compliant. A more detailed description of the CADET is

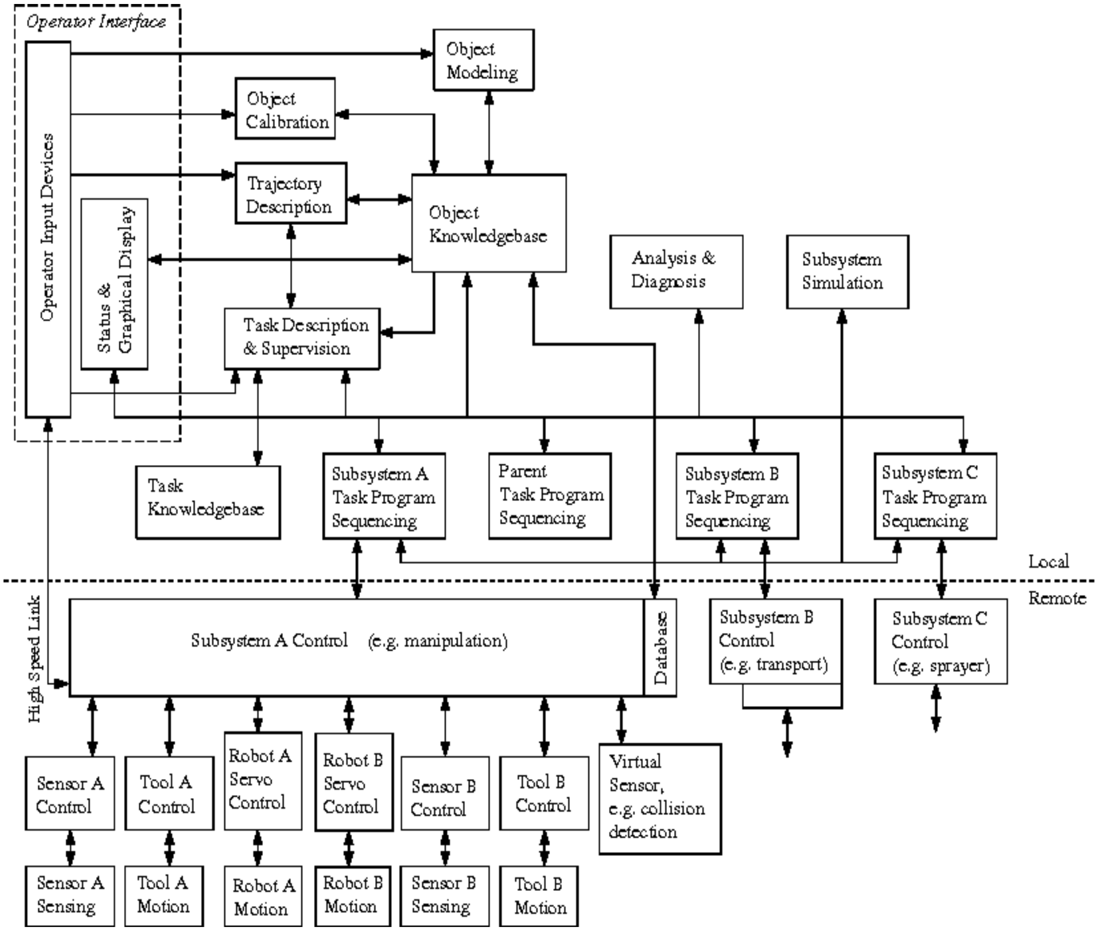

Figure 2: Telerobot Architecture for aircraft maintenance and remanufacturing

in Engel et. al. (4). The robotic arm is a Cincinnati Milacron T3-646 six axis electric robot. Communication to the robot is done via a serial port in which positions are sent every 60ms. Both controllers run on 68040 processors on a VME platform.

4.3 Software

The ADACS makes use of various off the shelf software packages. Only products that supported openness were selected. In general, a product provided openness by offering hooks to its functionality that allowed modifications or enhancements. The first of the products is Trellis NOMAD. NOMAD is a package of software modules that assist in making robot controllers. TMOS or Trellis MotionSystem is one particular element of the package. TMOS comes with a high level C interface that provides general motion control of robots. The TMOS is used for the ADACS trajectory generation. It is designed to be a part of an open system in that it is portable and interoperable with other components of a system.

ProManufacture, a module that runs within ProEngineer, is used to create tool paths. This data is run through a post-processor that creates tag points for use within TeleGRIP, the simulation software. TeleGRIP has hooks into it that allow customizing of the software. ADACS uses these hooks to specify tool parameters in each robot path when planning motion for a part. Once the edges have been selected to the users satisfaction, a part program is produced as output from TeleGRIP.

The workcell and robot controller runs on a 68040 processor running in a VME backplane. The operating system used is the LynxOS real-time operating system. This was chosen because it is a POSIX compliant operating system and allows for portability across different platforms. The Nomad motion control software was used for generating trajectories for the robot.

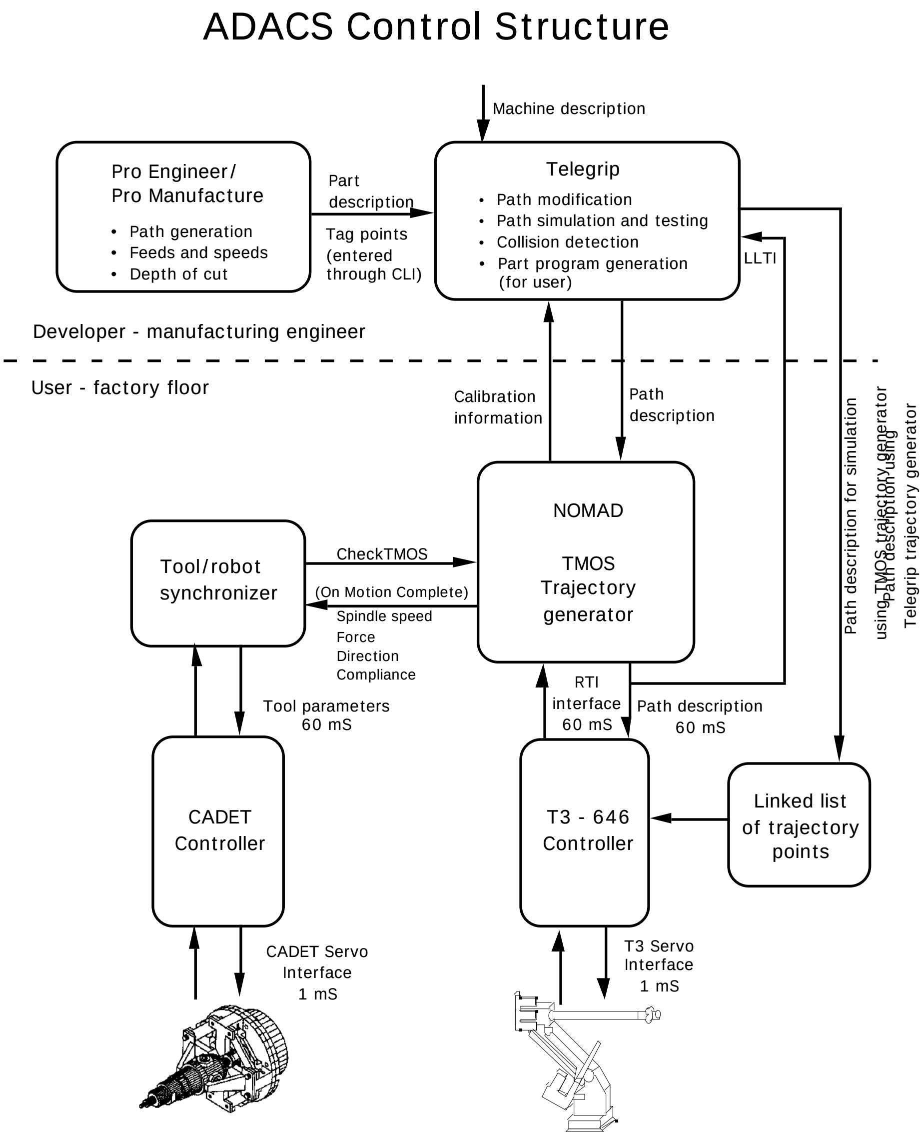

Figure 3: ADACS Control Structure

The CADET controller also runs on a 68040 processor in a VME backplane, but uses the VxWorks real-time operating system. The CADET servo level control software was developed at UTRC.

4.4 Integration

The integration of these modules described above is done by placing wrappers around the software that accept commands that are specified in the UTAP Interface Document by Lumia et al (5). By adhering to the interface standards specified in UTAP, future components with greater functionality developed by commercial vendors can be substituted for current components. The present state of control system integration is stymied by the lack of standards, an its is hoped that the UTAP architecture and interfaces can evolve into a standard.

Within the UTAP architecture, the description of module functionality and a detailed interface definition for each module is explained within the UTAP Interface document (5). This document assumes that the functionality of a module will satisfy the interface specification but the document does not mandate the program or algorithm used within the module. Since standard interfaces do not exist, middleware was developed to map each vendor-specific interface to the UTAP interface specification.

To connect modules within the system, an infrastructure must exist to provide common services for all component pieces. At NIST, the infrastructure consists of a set of tools that have been developed for module initialization and configuration, communication between software modules, synchronization, as well as other system services. As an example of a tool, a communication library is available that supports the use of either sockets and/or shared memory across many platforms. To handle module initialization and configuration, each module references an ".ini" files that defines its execution flags, resource allocation, communication connections and parametric settings.

5 CONCLUSION

The design and development of the ADACS NC Finishing System is described. Building an ADACS system from the ground-up was not cost-effective nor was modifying an existing closed-controller since such controllers are not sophisticated enough to provide the cohesive, real-time coordination of force-controlled tooling and motion control required for NCFS. Instead, ADACS was built using off-the-shelf hardware and software components and offered a challenge in system integration.

To facilitate system integration, the application of the Unified Telerobotic Architecture Project (UTAP) for the design of ADACS is described. Based on integration results, ADACS demonstrated that open architectures are useful in designing an NCFS. Real-time part finishing including coordination of multi-vendor software components did chamfer airplane parts. We found that a robust heterogeneous and distributed system infrastructure and set of integration tools - including common data structures, component naming, Interprocess Communication network communication, etc. - is imperative to facilitate component integration. To satisfy the UTAP interface standard, in-house middleware was developed as a programming front-end to the component. Applying this integration methodology, we would conclude that if you factor in software reusability, then this design approach is cost-effective and would be most useful in scaling the cost and performance to suit other NCFS applications.

As beneficial as openness is, one must note that openness alone does not produce the primary objective of NIST - standards. Openness provides benefits and savings through flexibility and extensibility but does not address portability. Interfaces under one vendor's open architecture generally will not run under another vendor's system. Openness is the first step towards standardization. The step after openness is to gather a consensus of opinion to then define a standard open architecture - as well as interfaces - that ultimately yields a standard open solution. In this regard, the ADACS project will serve as a testbed in an effort to design, test and disseminate open architecture solutions to finishing problems within the manufacturing industry.

6 References

1. Stouffer, K.A., Michaloski, J.L., Russell, R.T., Proctor, F.M., "ADACS - An Automated System for Part Finishing", from Proceeding of IECON'93, Lahaina, Maui, HI, November 15-19, 1993.

2. NASA JPL, "Final Report: A Generic Telerobotics Architecture for C-5 Industrial Processes", Air Force Material Command (AFMC), Robotics Center of Excellence (RACE), San Antonio Air Logistics Center, Kelley AFB, TX 78241, 1993.

3. Murphy, K.N., Proctor, F.M., "An Advanced Deburring and Chamfering System", Proceedings of Third International Symposium on Robotics and Manufacturing, British, Columbia, Canada, July 1990.

4. Engel, T. W., Roberts, R. K., Proctor, F. M., "Specification of an Active Force Control Tool for Performing Deburring and Chamfering on a Robot Platform", Proceedings of the International Conference on Industrial Electronics, Control, Instrumentation and Automation, November 1992, pp. 918-926.

5. Lumia, R., Michaloski, J.L., Russell, R.T., Wheatley, T.E., "Unified Telerobotic Architecture Project (UTAP) Interface Environment", NISTIR 5658, National Institute of Standards and Technology, Gaithersburg, MD, 20899, May 1995.