Automation and Robotics in Construction XI
D.A. Chamberlain (Editor)

Design for automation - A Generic System Architecture for Civil Engineering Applications

A Middlebrook

National Advanced Robotics Research Centre, Salford, Gtr. Manchester, England.

ABSTRACT

The objective of ESPRIT III project n. 6660 - RoadRobot - Operator Assisted Mobile Road Robot for Heavy Duty Civil Engineering Applications, is the development of a Generic Multipurpose Control Architecture which will be applied to an outdoor construction site consisting of several heavy duty platforms. One of ARRL's functions within the project was to create the "template" by which a real construction site may be created. This paper briefly describes the Functional Architecture, which is a description of the LOGICAL resources which are used to produce system outputs and information flows between Resources.

1. INTRODUCTION

The work presented in the following paper has been done within the ESPRIT III Project No. 6660 - RoadRobot - Operator Assisted Mobile Road Robot for Heavy Duty Civil Engineering Applications. This project consortium is made up of several European partners, some academic, some potential end users, but all with an interest in the design and development of an automated construction site.

The Architecture described is based on the overall approach developed by all of the partners during the initial proposal writing phase and is linked closely with concepts of Computer Integrated Manufacturing (CIM) to incorporate improvements in productivity, flexibility and quality to the construction site.

The main objectives when considering the architecture were to produce a general architecture which would facilitate the introduction of automation into the construction site and would follow the following principles.

* Object Oriented

* Hierarchical Design

* The architecture should allow for gradual implementation.

* Have strong links with other IT construction approaches

Design of the architecture was done by employing the Shlaer-Mellor methodology ①.

The RoadRobot architecture can be thought of as a "template" by which a real construction site may be created. The architecture defines objects and categories (ENTITIES) that have the possibility of existing within a construction site. The relationship that these entities have with each other is also defined. By analysing the ways in which current construction sites exist and how work is planned and executed it is possible to form such a template.

It is a requirement that current construction sites must fit within the architecture and be able to integrate new technologies and by doing so gradually increase the number of functions that will be taken on by the autonomous machines and intelligent planning systems. It is also necessary to include various "levels" of autonomy since it is recognised that the knowledge initially acquired will require a great deal of refinement.

2. ARCHITECTURAL HIERARCHY

The architecture described later is a generalised Functional Architecture and as such is only one of a number of different descriptions. Within the area of Robot architecture there exists several different levels of abstraction and various generic issues can be handled at each level.

2.1 The Implementation Architecture

The Hardware and Software Architecture are the body of the Implementation Architecture for a computer based system and together describe the way in which an actual system design has been implemented for a specific robot device. The Hardware Architecture describes the design and interconnection of the data processing devices and the Software Architecture refers to the algorithms and their relationship to data structures.

2.2 The Operational Architecture

The Operational Architecture is concerned with the specific operational capability of a robot system and is posited at the level of a virtual machine. Here the system appears as a series of black boxes which given inputs achieve specific outputs. Emphasis at this level is placed upon functions implemented and data flows or command language between modules. With this level describing multiple virtual mechines, an operational architecture could be designed for a series of mobile vehicles with similar capabilities and changes within one virtual machine do not affect the other virtual machines. Porting between applications (vehicles in this case) can be achieved through implementation level changes in machines that are affected by the changed hardware.

Mapping from the Operational Architecture to the Implementational Architecture can be a complex task and potentially involves both one to many and many to one mappings.

2.3 The Functional Architecture

This is a description of the LOGICAL resources which are used to produce the system outputs and the information flows between those resources. A resource would typically be described in a prototypical form at this level. The functional architecture has a great potential for generic applications, this initially complex system gives rise to different implementation architecture for different applications.Mapping from the Functional Architecture to the Operational Architecture may also involve one to many and many to one mappings (e.g. a single Physical Sensor may be used for the implementation of a Logical Sensor performing collision detection and another object recognition). It is this level that is briefly described in the following sections. All the following diagrams are a not using the methodology chosen, but are a representation of key ENTITIES.

3. LOGICAL LEVELS

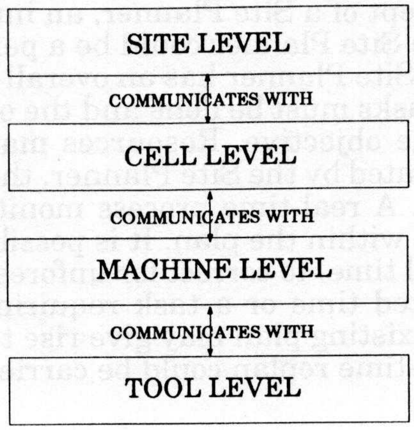

Figure 1.
Hierarchical Schematic

The following hierarchical logical levels (see Figure 1.) are defined within the Generic Functional Architecture of RoadRobot ②. The levels deliberately share as many feature with each other as possible. All four levels for instance are coordinated by a level planner. This planner, the Controller, may either be a person (as it is in current existing building sites) or an intelligent computer system (the ideal objective of RoadRobot).

In each case the Controller may be aided by an Operator if it cannot reliably form correct plans from its knowledge base and inputs.

3.1 The Site Level

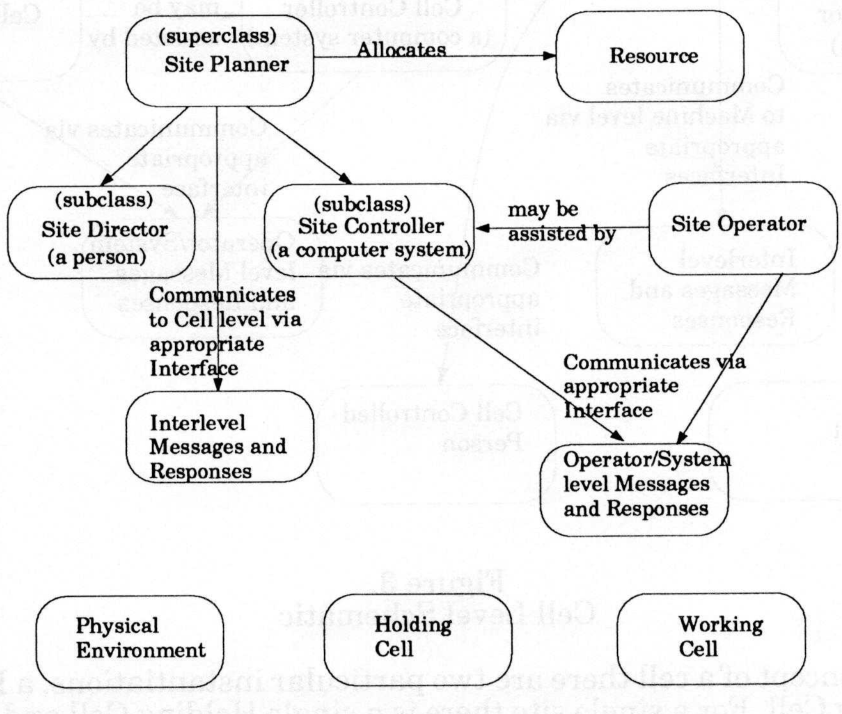

Figure 2.
Site Level Schematic

The Site Level (which can be thought of as the top level in the architecture) is the module that effectively overviews the activities of the construction site. This level centres around the concept of a Site Planner, an intelligent planning system (a non real-time system). The Site Planner could be a person (its existing form) or some computing system. The Site Planner has an overall goal(e.g. the construction of a road) and decides what tasks must be done and the order in which these tasks must be done, to achieve the objective. Resources may then be allocated to a conceptual cell which is generated by the Site Planner, the cells being used to carry out steps of the overall plan. A real-time process monitors the Cells in order to detect the completion of steps within the plan. It is possible for the Site Planner to modify its overall plan (in real time) to correct for unforeseen circumstances (e.g. a task running over its allocated time or a task requiring extra Consumable). A real-time modification to an existing plan may give rise to a non-optimal plan, but if it was so desired a non real-time replan could be carried out.

3.2 The Cell Level

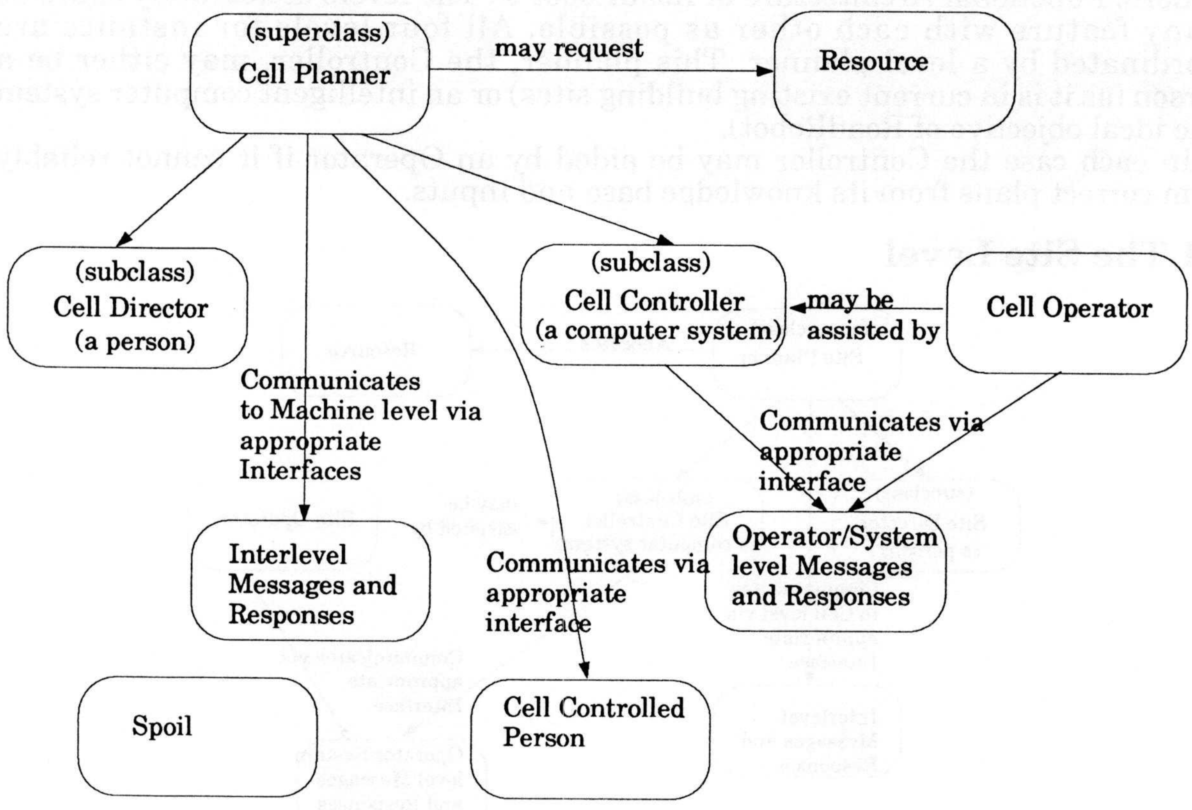

Figure 3.
Cell Level Schematic

Within the concept of a cell there are two particular instantiations, a Holding Cell and a Working Cell. For a single site there is a single Holding Cell and one or more Working Cells. As with Site Level , the Cell Level is based around a Cell Planner (which may either be a Cell Controller or a Cell Director). The Site's perception of the needs of the Cell to perform its task reflects the allocation of resources. The architecture also allows for the possibility of a Cell to request resources that it believes it needs. This allows for the possibility of more flexible overall plans to be made (e.g. sharing a rare machine between Cells that have an infrequent requirement for that machine). The Cell Planner produces a detailed temporal plan of the given task e.g. in order to achieve the task of laying a strip of road many passes of the road paver and its associated support machinery would be required. This planner must be as close to a real-time planner as possible and in order to achieve this it draws heavily on prior and pre-planned temporal plans.

The Holding Cell's task is to maintain currently unallocated resources that are available to the site. Major maintenance is taken care of here, as is the delivery of external resources.

The Working Cell's task is to carry out work upon the Site. A typical example of a Working Cell might contain an excavator, several support machines, a number of operators and enough material to perform the task of filling a trench.

3.3 The Machine Level

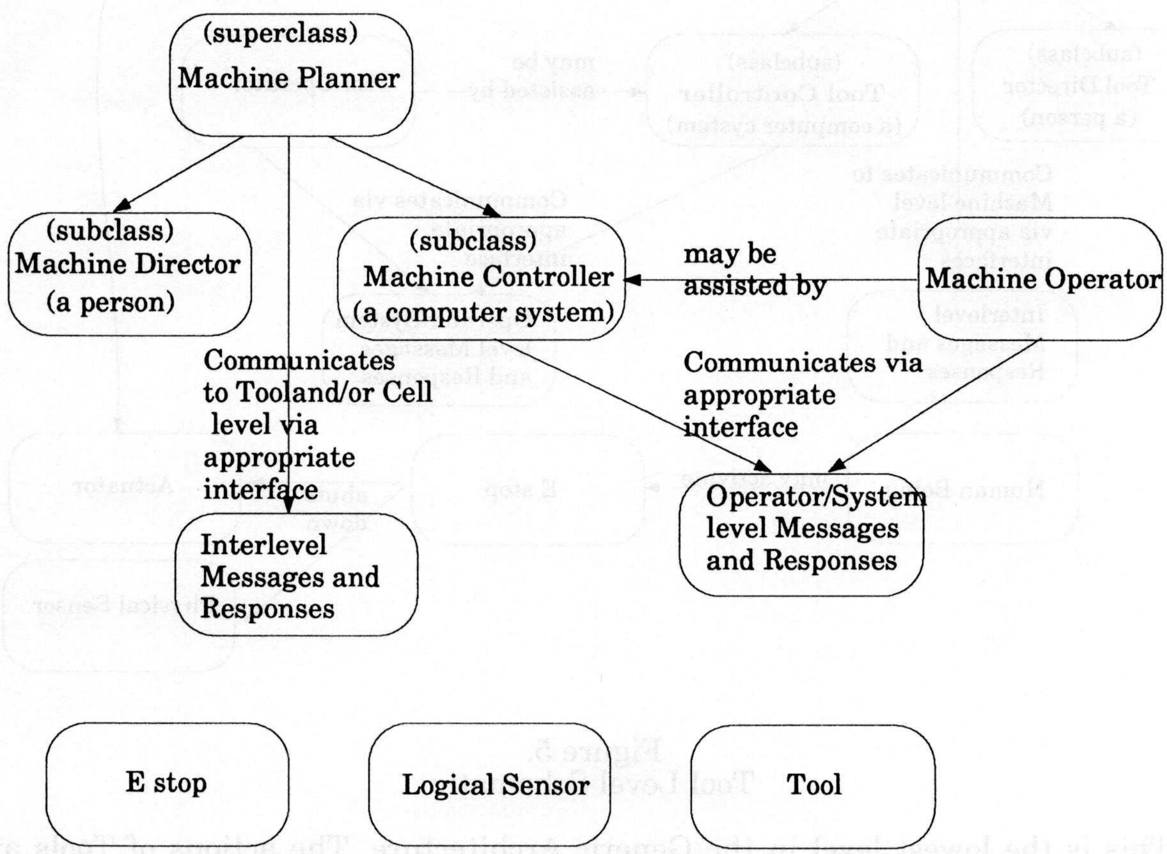

Figure 4.
Machine Level Schematic

At Machine Level there too is such a thing as a planner - a Machine Planner, which may be either a Machine Controller or a Machine Director. At Machine Level, low level control of the system starts. Here the Machine Director may be a driver with communication with the Cell level, the Machine Controller being a computer controlled system. The Machine Planner (whether Controller or Director) using its local knowledge and taking a task prepared for it by the Cell Planner, then coordinates its Tools to perform the task.

In order for useful decisions to be made, the Machine Planner is required to produce plans in real-time. This level also incorporates Logical Sensors which reads values of Physical Sensors and if appropriate may fuse the data from the Physical Sensors into information that may be useful for the Machine Planner.

3.4 The Tool Level

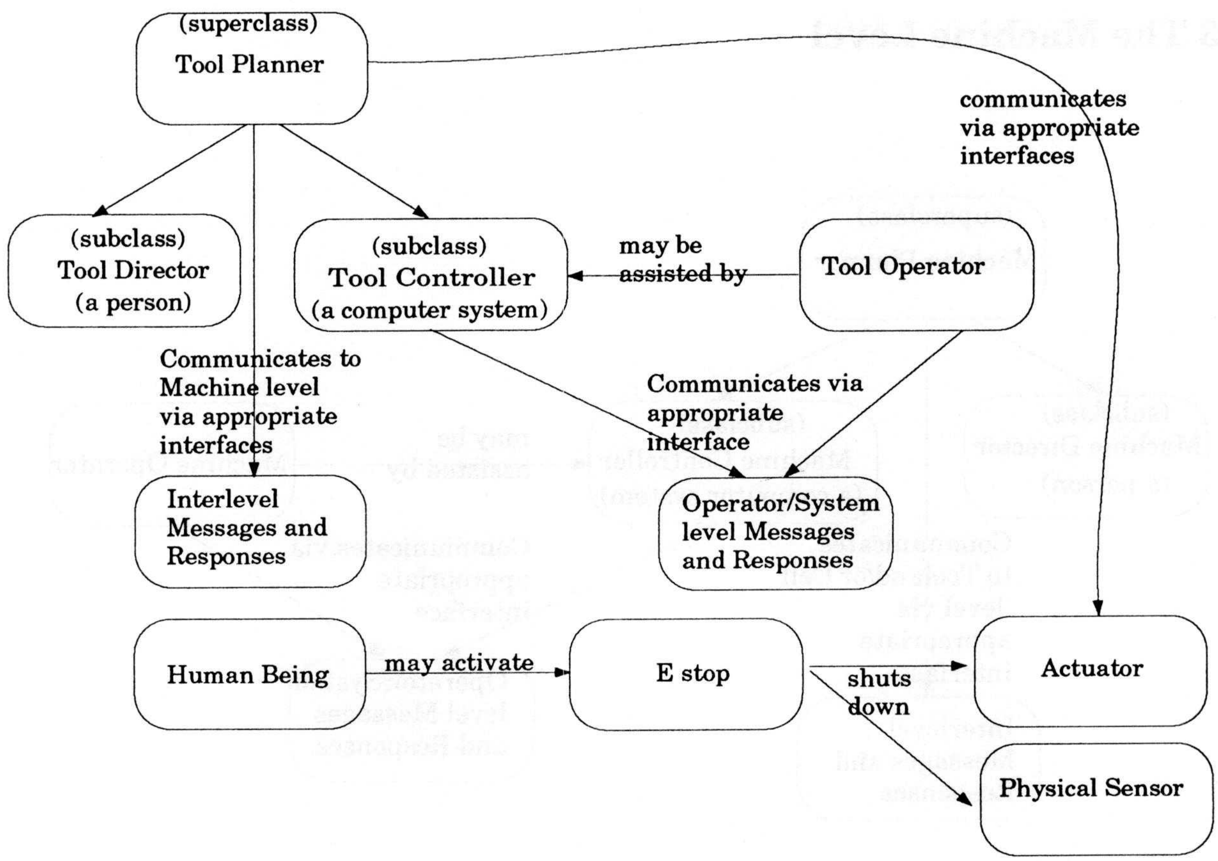

Figure 5.
Tool Level Schematic

This is the lowest level in the Generic Architecture. The actions of Tools are coordinated by Tool Planners which must run in real time (since it is the low level control loop of the system). Incorporated within a Tool there may be many Actuators and Sensors that are used in order to affect the physical environment in a controlled manner.

4. OTHER AREAS

4.1 Human Beings

Many different Types of Human Beings are considered within the Generic Architecture. It was decided to restrict the range of decisions/roles a person could carry out at any one time. The restriction clarifies the question of which Human Being is allowed to command/plan, and also the ENTITIES that they may instruct with these commands.

The highest level of person, a Human Being, is anyone in connection with the site, and all have the ability to activate an E-stop at any time. Sub-types of Cell Person and NonCell Person are used to differentiate the various types of people that a Cell may incorporate/control. Cell Persons are then further subdivided into Controlled Cell Persons (who communicate directly with the Cell Planner) and NonControlled Cell Persons who receive instructions from Controlled Cell Persons. Administrators are people who are directly involved with each logical level and are split into Directors (people who actually plan for this level - analogous to people who perform these functions on construction sites) and Operators who are able to "aid" the various computer controlled systems that exist within the concept.

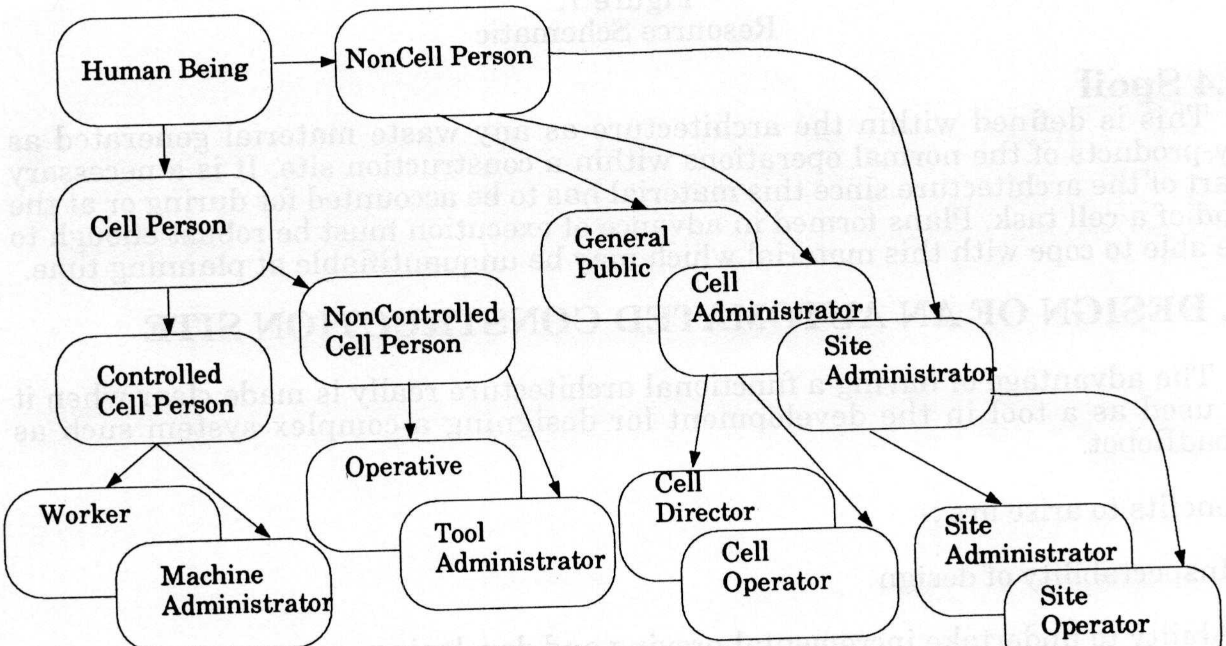

Figure 6.
Human Being Schematic

4.2 Communications and Interfaces

Any communication between entities is carried out using two associated entities (e.g. between the Site and Cell entities there would exist a Site_Cell_Interface and a Cell_Site_Interface and between Site and Operator a Site_Operator_Interface and Operator_Site_IF. Information passed "downwards" through the system are called Messages and "upwards" Responses (e.g. Site_To_Cell_Message and Cell_To_Site_Response). Information may either be solicited or unsolicited. Tasks may only be passed down within Messages however.

4.3 Resources

A Resource is defined within the architecture as some physical entity that is required by a cell for the completion of its task. This includes entities such as Machines (intelligent and unintelligent), a Cell Controller, Workers and any Consumables. Unused current Resources (and new Resource deliveries) are overseen by a Holding Cell.

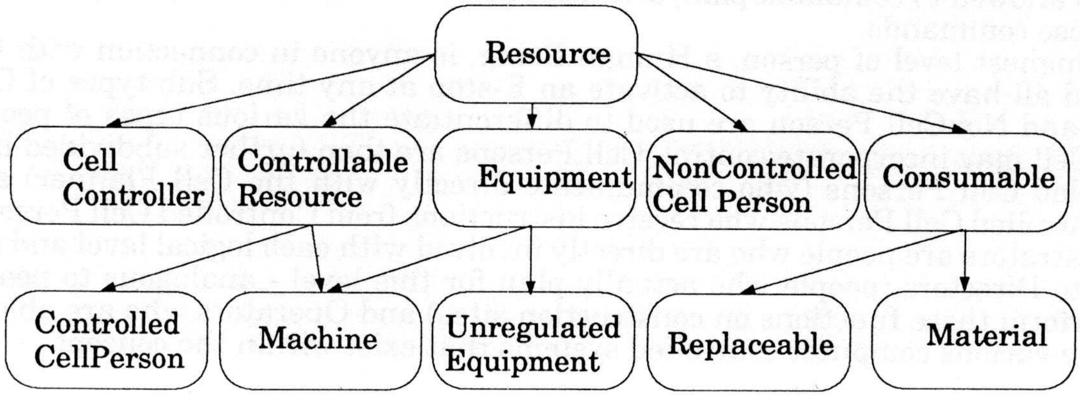

Figure 7.
Resource Schematic

4.4 Spoil

This is defined within the architecture as any waste material generated as by-products of the normal operations within a construction site. It is a necessary part of the architecture since this material has to be accounted for during or at the end of a cell task. Plans formed in advance of execution must be robust enough to be able to cope with this material which may be unquantifiable at planning time.

5. DESIGN OF AN AUTOMATED CONSTRUCTION SITE

The advantage of having a functional architecture really is made clear when it is used as a tool in the development for designing a complex system such as RoadRobot.

Benefits to arise are :-

* Inspectability of design

* Ability to undertake incremental proving and developing

* Common design approach amongst different applications

* Partitioning of design amongst teams and individuals.

Having produced the functional architecture description, the next stage as with RoadRobot is to produce a functional definition of the interfaces between the entities. Following this a series of Operational Architectures are produced . One Operational Architecture is developed for each major type of Machine to be developed. In theory one could conceive of different Operational Architectures for different types of construction sites. Operational Machines relate to virtual machines and as such it is possible at operational level to describe say, all excavator types as one virtual machine with the same operational architecture. Certainly within RoadRobot all road pavers could be covered with a single architectural description.

An initial first pass instantiated architecture for a road paver has already been produced. As was expected during the instantiation process the architecture as well as becoming more specific, expands a great deal. This expansion arises from the need to detail each specific instance of details such as each tool on the machine, rather than just vaguely describing the category and its characteristics.

The next stage after this instantiation is to produce State Transition Diagrams which describe all of the possible states that an entity can exist in as well as the inputs related to those states. The input and output descriptions can in turn can be refined to become detailed interface definitions between each entity.

The "traditional"design phase of detailed hardware and software is then undertaken, this is based upon the Operational Architecture specifications. By following the constraints of the operational and interface definitions of the Operational Architecture problems are minimised during the integration phase. However it must be stressed that deviations in performance or interface specifications during project development should be assessed.

6. CONCLUSION

The architecture is now being used by all members of the consortium as an aid in the design and development of modules associated in the automation of a heavy construction vehicle. The final physical demonstrator will be that of a road paver, and simulations will be carried out for that of an excavator. The architecture is proving a very useful tool in the sense that all partners recognise it almost as an interface between design tasks which can often be dissimilar. It also highlights areas which may have been overlooked in other design processes. The fact that the architecture presented is being used does not make it ideal and there could well be situations which the architecture does not encompass, provision has been made for minor alterations at a later stage. The paper however, presents the whole design process of RoadRobot as an entirely linear affair. In reality there are numerous 'loop backs' and interactions between stages, but the overall process, described briefly, presents the approach which will be followed.

References

1. S Shlaer, S. Mellor, Object-Oriented Systems Analysis: Modeling the World in Data (Prentice Hall, 1988).
2. A. Middlebrook, I. King, ESPRIT PROJECT 6660, The Generic System Architecture (1993).