# Standardised Framework for Flexible Materials Handling Management based on Operating System Primatives

**Glen Bright and Anthony Walker**
Mechatronics and Robotics Research Group
Department of Mechanical Engineering, University of KwaZulu-Natal,
Durban, South Africa.
203507919@ukzn.ac.za

## Abstract

Future Flexible Materials Handling systems will employ multiple cooperating mobile materials handling platforms in order to carry out required materials handling tasks. Mobile robotic system research and development is by nature, distributed. An internationally developed and accepted Flexible Materials Handling management framework will aid in correlating research efforts and allow for system developments to be communicated across a standardised and well understood platform. This paper presents a layout of such a standardised framework, structured around the model of a multi-user Operating System in order to develop well understood and generic mechanisms. This aids in the realisation of Flexible Materials Handling (FMH) systems.

## 1 Introduction

Mass production of customised products requires materials handling systems which are flexible and responsive enough to accomodate real time changes in material payload configuration and characteristics. A final report on a project in the Intelligent Manufacturing Systems (IMS) program describes the characteristics which a materials handling system should posses in order to accomodate the materials handling requirements of modern manufacturing systems. These characteristics are listed here accompanied by short descriptions [MMHS Metamorphic Materials Handling System, 2003].

- Responsive: The materials handling system should be responsive to changes that may take place in manufacturing technology, type of product or material to be handled, work schedule and load.
- Flexible: The materials handling system should be capable of transforming itself and altering its function to meet any changes in handling requirements.
- Autonomous: Able to make decisions on its own and function as closed system during materials handling task instances.
- Highly Automated: Incorporated with next generation automation technology.
- Multi-functional: The materials handling system should provide functions such as assembling, packaging and disassembling besides transportation.
- Modularised: The materials handling system should be composed of various modules, each with a distinct function such as a planning module and a communication module thus providing the platforms with the capability of associating and dissociating one another to organise themselves into a required configuration.

From the above listed characteristics it is evident that management of such materials handling systems has to be layered and hierarchal in order to cope with the inherent complexity of such systems.

Although distributed autonomous materials handling platforms are required in order to carry out the physical materials handling tasks, these systems themselves can not solve the problem of Flexible Materials Handling and support the mass production of customised products. The solution will require a high level management system which is well integrated into the manufacturing system and machine tool infrastructure and encapsulate the scheduling and task allocation mechanisms required to sustain production momentum. Manufacturing systems operate at a dynamic in accordance to the product processing operations taking place and the machine tool charateristics and performance. Task allocation systems should be able to utilise this dynamic in order to maximise material and product diffusion through the manufacturing system thus maximising efficiency and product output. Without this information, quantification of system performance would not be possible thus preventing any means of system tuning and optimisation. Machine tools and other manufacturing infrastructure produce temporal characteristics analogous to computer peripherals such as hard drives and CD-ROM's. In these instances the Operating Systems kernel schedules processes in such a way as to prevent mechanical latency from reducing system performace by scheduling other processes for CPU usage whilst the hard drive or CD ROM is performing its distributed operation. The same scheduling procedures can be employed to maximise manufacturing activity by scheduling materials destined for machine tools which have higher operation bandwidths more often. In UNIX Operating System terminology, a runlevel is a mode of operation describing the operating state of the Operating System. The runlevel concept can be applied to Flexible Materials Handling by placing groups of materials handling tasks into runlevels and scheduling resources accordingly. Operating Systems utilise system calls to allow user space applications a generic method for requesting services from the kernel.

Means should be provided to allow for the manufacturing management system to request materials handling services through well defined and standardised system call interfaces. The system call interfaces are discussed in section 2. Assembly machine tools can produce situations which are analogous to critical sections where the initiation of the assembly operations requires consecutive component delivery to the machine. The Flexible Materials Handling system should be able to perform the consecutive materials handling operations whilst holding a semaphore and release the semaphore on exiting the critical section. Developments in Computer Science have shown that complex problems are transformed into sane equivalents by breaking down the problem into a well organised hierarchy. The same transformation can only aid in the structuring of modern manufacturing systems and the Flexible Materials Handling systems which will supply material flow through the manufacturing system.

The objectives of this research were to:

- Develop a framework in which future Flexible Materials Handling sytems could operate.
- Develop international interest in the development of standardised operating mechanisms which would allow for correlated research efforts.
- Learn from the principles of Computer Science and Operating Systems to model the framework on a commonly understood platform.

## 2 System Structure and Philosophy

The internal mechanisms and governing concepts of the framework presented here are derived from Operating System primitives. Operating Systems provide virtual abstractions of the underlying physical machine. They handle the allocation of resources to running processes and provide clean generic structures that shield the technical aspects of system operation from the machine operator.

The framework developed here uses the concepts of "kernel space" and "user space" to distinguish between the limited resource enviroment of the physical materials handling platforms and the unlimited resource environment of the Database Management Systems presenting customised product configurations, dicussed in subsection 2.1.

The system is generic and modular allowing for distributed development of the system constituents. This framework is intended to be generic enough for implementation in a variety of manufacturing environments, with only the parameter structures needing adjustiment. A top down approach was used during the development of the framework and starts from customer specifications working its way down to the physical platforms performing the materials handling operations. Discussion of the system structure is split into three subsections. The User Space, the System Call Interface and the Kernel Space.

### 2.1 User Space

The User Space functional blocks of the Flexible Materials Handling management framework map customer specifications entering the manufacturing plant to well defined and detailed materials handling tasks. The customers desired configuration is passed through a simulator to verify feasibility of the request, simulation will play an important role in future manufacturing systems. Current technologies utilise Computer Aided Process Planning (CAPP) systems to formalise structured manufacturing operation sequences from customer specifications. In this light, these systems would be part of the User Space and exist in the context of their particular specialist field. Figure. 1. shows the main components of the User Space environment.

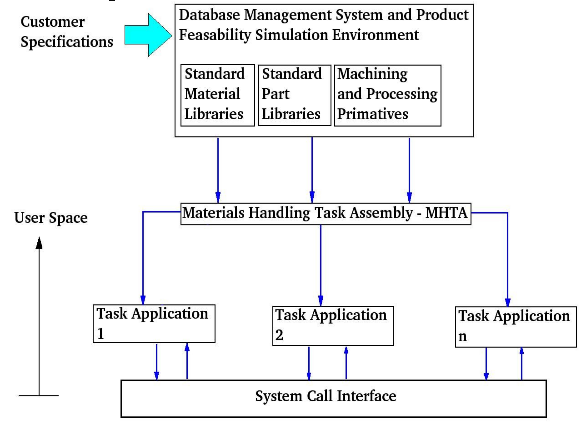


Figure 1. User Space environment.

The User Space environment is not directly linked to the scheduling and allocation of physical materials handling operations although a short discussion is included here to highlight the benefits of such an approach to materials handling management.

The customer is presented with a Database of product components and subsystems. The customer chooses a desired product configuration. This could be performed on a web site for example or through direct company sales departments. The configuration is then verified through the use of manufacturing simulations. The simulations make use of standardised part libraries and machining primitives. These components can be thought of shared libraries which are linked to the simulation executable for a particular manufacturing environment and product configuration.

The infomation is then fused into materials handling tasks in the Material Handling Task Assembly unit. The materials handling tasks are described by task applications which contain all relevant information as to the sources and destinations of the materials around the manufacturing plant, the volume of materials requiring transportation, material parameter structures for precise material specifics, the number of materials handling platforms required to carry out the tasks, the material handling task priority etc. These task applications interface with generic system calls. There are many benefits to modelling Flexible Materials Handling systems on Operating System primatives. These include:

- Allowing for international cooperation in the development of generic system calls which can cater for the different materials handling tasks.
- Allowing everyone to communicate on a common platform which can increase the development rate of such systems and prevent uncorrelated development.
- Allowing for the structuring of mechanism and policy, two important aspects of future manufacturing systems.
- Integrating the machine tool dynamics into the task scheduling systems allowing for higher material diffusion through the manufacturing plant.

The User Space environment can be developed separately from the kernel subsystem as long as it interfaces with generic system calls.

### 2.2 System Call Interface

The User Space task applications developed from customer specifications require a generic method of requesting materials handling services form the materials handling infrastructure. For this reason a standardised set of system calls need development to standardise the way in which Flexible Materials Handling platforms are called into operation. Systems of multiple cooperating mobile platforms are under development to increase materials handling scope in order to accomodate varying payload geometries and characteristics. System calls which describe the required materials handling operation in terms of the number of platforms required is a good starting point and was considered in our framework development approach. Manufacturing systems producing mass customised products need a layout structure which incorporates port structures from which the materials handling platforms can collect and deliver material payloads. These port structures are depicted in Figure. 2.

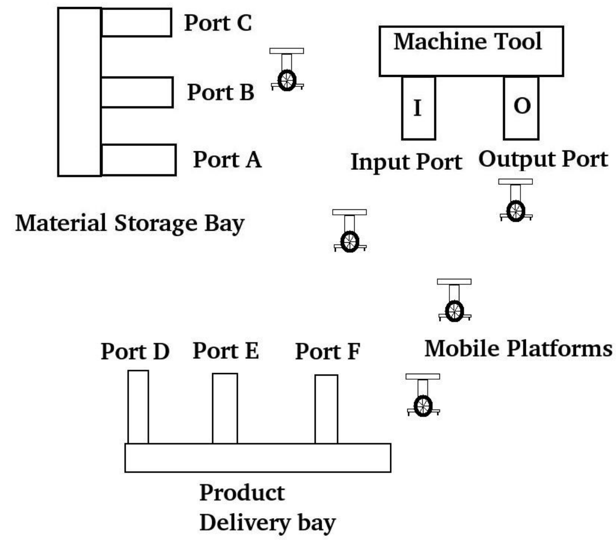


Figure. 2. Port strucures in manufacturing environment set out in producing mass customised products.

In order for a task application to gain access to the materials handling infrastructure it must open an active task instance on the required port structure. A port structure could be anything from a warehouse dispatching conveyor to an output conveyor fitted onto a machine tool. This is made through the "open" System Call as shown in Figure. 3.

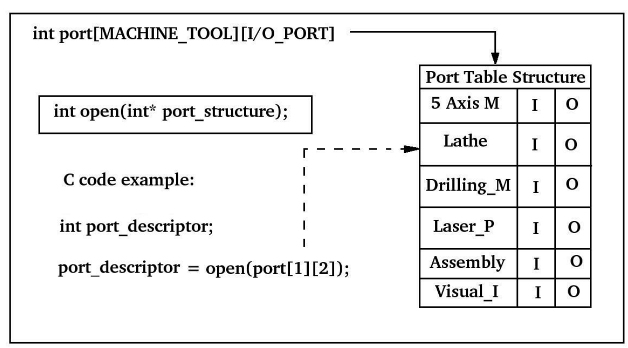


Figure. 3. Diagramatic layout of the open System Call.

In order for the system to operate, a port table must be constructed in user space and is manufacturing system dependant but generic in the sence of its purpose.

The open System Call returns a task descriptor to the calling task application in user space. This represents the unique connection which the task application has with the port infrastructure of the manufacturing environment. This System Call is directly derived from the "open" System Call of UNIX-like Operating Systems.

The calling task application in user space will then pass in the task descriptor to the System Call which initiates a physical materials handling task from that port structure.

Figure 4. shows a diagramatic layout of a System Call which manages the request of a multiple platform materials handling task.

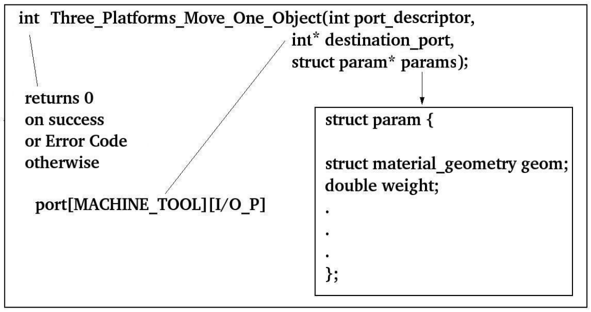


Figure 4. A Flexible Materials Handling System Call

The System Call is made from the User Space task application and parameters are passed in describing the material destination port and the parameters of the material to be transported. The parameter structure allows the participating mobile platforms to develop actuation primatives to load and unload the material at the end points of the materials handling task. The exercise of developing actuation primatives is called the Parameter Interpretation Method (PIM). The PIM is a hardware dependant entity and can be written to map parameters to the mechanical and electro-mechanical actuation primatives for a particular platform architecture without breaking down the User Space parameter structure.

## 2.3 Kernel Space

The Kernel is the most crucial part of an Operating System and interacts directly with all the low level programmable elements included in the hardware platform.[Bovet, D. P and Cesati, M. ,2000].

In the context of the framework presented here the Kernel space mechanisms perform most of the work in realising physical materials handling tasks from User Space task applications. The Kernel performs the scheduling of material handling task instances, resource management, plant status monitoring, task allocation and runlevel switching. Other mechanisms running in Kernel Space include a locking mechanism which allows task instances to hold particular port structures during “critical sections”. An instance of a critical section can be during a material transfer to an assembly machine tool. The machine tool requires consecutive material delivery to initialise the assembly operation. If another material handling task instance uses the same port structure, the assembly operation would fail and cause unnecessary down time.

An outlay of the Kernel space mechanisms is shown in Figure. 5.

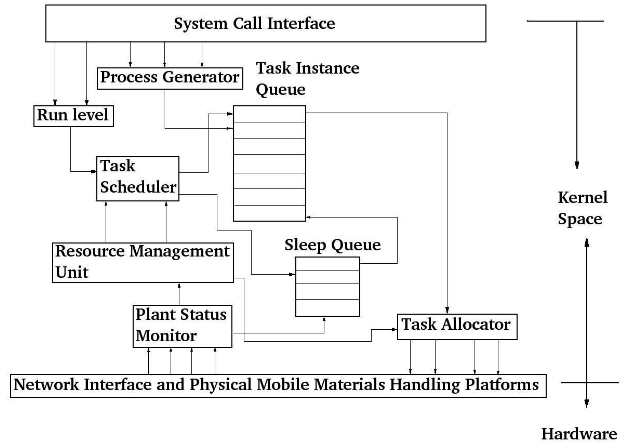


Figure. 5. Kernel Space environment

### The Runlevel

The runlevel of the materials handling task is determined by three factors. The volume of material requiring transportation on a particular task descriptor, the processing bandwidth of the destined machine tool and the platform mode of the materials handling task. The runlevel determines how a particular task instance is scheduled for operation. Lower runlevels recieve scheduling precedence. The mode of the materials handling task is a description of the kind of materials handling whether it be multiple cooperating mobile platforms or single mobile platforms.

### Process Generator

From now onwards a process is equivalent to a task instance of a particular task application. The process generator maps a particular system call on a task discriptor to a process structure which can be interpreted by the task allocator.

### The Task Scheduler

This unit schedules processes according to their runlevel and available resources. Scheduling algorithms can be developed which maximise manufacturing system performance. Algorithms to maximise material diffusion would aid in achieving these high performance measures.

### Resource Management Unit

The Resource Management Unit monitors the Plant Status Monitor and builds an effective resource collection to be used in scheduling algorithms.

### Plant Status Monitor

This kernel functional block monitors the status of all active machine tools and materials handling platforms to present raw status data upon which the Resource Management Unit can build resource collections.

**Task Allocator**

The Task Allocator maps processes onto physical materials handling platforms and passes on associated data structures, which were passed to the system calls as parameters, to the participating mobile materials handling platforms. These task structures are then interpreted by the mobile platforms via Parameter Interpretation Methods. The Task Allocator reads from the Resource Management Unit in order to choose the best suited mobile platforms based on the required tasks. This is the Optimal Assignment Problem (OAP) of Operations Research which has created much research in the distributed robotics community[Gerkey, B. P. and Mataric' M. J., 2004]. The conceptual task allocation mechanism presented here is based on the estimated power requirements of the materials handling operations and platforms are assigned to materials handling operations in such a way as to produce Stable Mode Convergence (SMC) of the Platform Power Capacity Distribution (PPCD). (shown in Figure. 6).

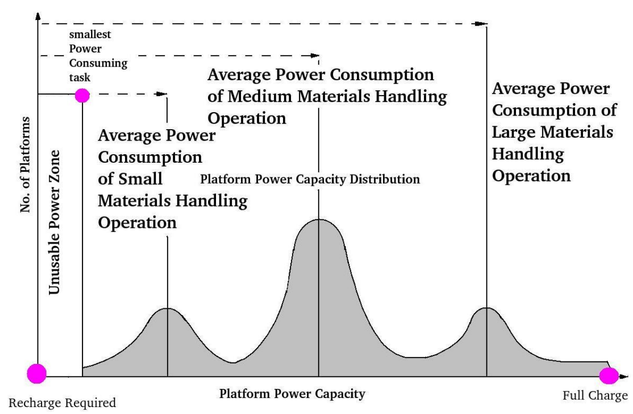


Figure. 6. Platform Power Capacity Distribution

This method of task assignment tries to maximise system performance by minimising the amount of operational mobile platforms with power resources in the Unusable Power Zone. The Unusable Power Zone represents the range of power capacities between a mobile platforms fully drained state and the power requirement in order to perform the least power consuming materials handling operation in a particular manufacturing environment. When the Task Allocator reads from the Resource Management Unit it will assign platforms to a given task instance based on their output mappings onto the PPCD. Platforms whose output mappings maintain the integrity or rather stability of the PPCD will be assigned to the task instance. An output mapping is simply the estimated power capacity of the platform after the materials handling task. Stable Mode Convergence of the PPCD means that the platforms are under maximum utilisation and therefore the materials handling system is at its maximum efficiency. Although this paper is concerned with the development of a Flexible Materials Handling framework, optimal task allocation will play an important role in the functioning and efficiency of Flexible Materials Handling systems in manufacturing environments mass producing custom products.

## 3 Generic Communication Structures

### 3.1 Robot Servers

In order for the kernel structue of the Flexible Materials Handling framework to remain generic, the use of Robot Servers will be required to handle the task allocations in a standardised format. This is depicted in a simpified diagram of a process execution through kernel space in Figure. 7.

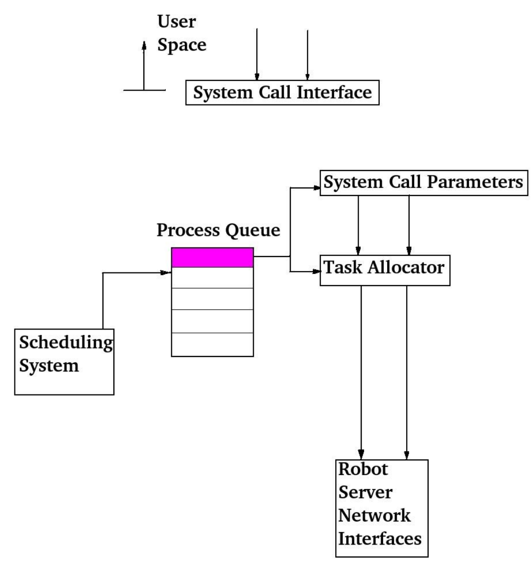


Figure. 7. Simplified process execution through kernel space.

Interface specifications need international development to produce a standardised format of task discriptions which can be allocated to mobile materials handling platforms through standard network interfaces. Plugin drivers can then be written to map the generic commands sent to the platforms over the network into specific Flexible Materials Handling operations and interpretations.

For Instance, the Parameter Interpretation Method would be one of these plugin drivers. The System Call depicted in Figure. 4. is used as an example. The parameter structure passed in to the system call containing relevent material information can be interpreted by the PIM driver running on a particular instance of the Robot Server running on the onboard computer of the mobile platform in context. The PIM driver would produce relevent actuation primatives for that particular materials handling operation. Such actuation primatives could be used for material configuration adjustment or during offloading and onloading of the materials at the end points of the Flexible Materials Handling task space.

Current research in the Mechatronics and Robotics Research Group $MR^2G$ is utilising the Player Robot Server[*Collett et* al., 2005]. This provides the Hardware Abstraction Layer (HAL) for the mobile robots involved in Flexible Materials Handling research.

Future research will be involved in integrating PIM interfaces into the Player Robot Server to allow for its participation in the devleopment of Flexible Materials Handling frameworks. The PIM interfaces will be used as binding mechanisms for plugin drivers.

## 4 Management System Placement

The placement of a Flexible Materials Handling management system should complement existing manufacturing management systems to provide a high level of integration. Current management and automation systems such as Siemens, Totally Integrated Automation (TIA)[www.siemens.com/totally-integrated-automation], provide management and distributed control from the upper levels of resource planning through to plant control and Fieldbus technologies. The FMH management system proposed in this paper is a likely candidate for integration into such frameworks. The closest subsystem correlation of the existing Siemens Totally Integrated Automation system and the system proposed in this paper is the Manufacturing Execution Systems [www.siemens.com/totally-integrated-automation]. Figure 8 shows the Flexible Materials Handling management system in the context of network infrastructure.

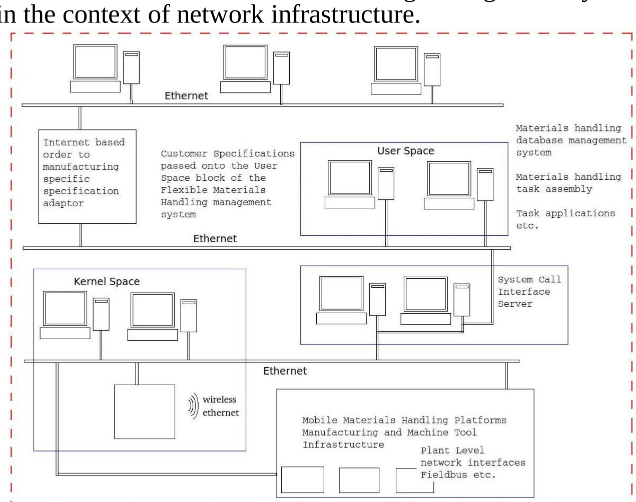


Figure 8. Placement of Flexible Materials Handling management system in a network structure.

Figure 8. is a representation of a particular orientaion to materials and equipment management in manufacturing environments set out in producing mass customised products. Current industrial networking structures are suitable foundations for all levels of manufacturing management, process management and hardware management. In manufacturing environments mass producing custom products, the modular layout of FMH management system will allow for an integrated execution environment with high production momentum whilst still supporting the real-time nature of mass customisation.

## 5 Discussion

Developing a standardised Flexible Materials Handling management framework provides a common platform for international development of next generation Flexible Materials Handling systems.

Development of functional mechanisms such as the Task Allocator in the context of a kernel object provides a common international interpretation of the Task Allocator itself. This allows for the development of optimal assignment algorithms housed in a standardised structure applicable to all Flexible Materials Handling systems based on the same international framework.

Development of the framework subsystems can be communicated globally in a common format producing greater global development rates in manufacturing industries set out in producing mass customised products.

The initiation of the Flexible Materials Handling management framework on an international scale will require the development of task groups which are explicitly involved in one aspect of the framework. International task groups can be assigned to User Space development, other international task groups can formalise the System Call interface specifications whilst other task groups develop the kernel space subsystems, scheduling algorithms and optimal assignment architectures.

The framework allows for the explicit integration of machine tool dynamics into the materials handling system and provides a hierarchy in order to break down the complexity of the Flexible Materials Handling system into managable levels of complexity and detail. The networking hardware and software present in modern manufacturing systems aids in optimising system performance whilst concentrating pivotal information into unified communication mediums. The configuration of such information frameworks in the context of FMH requires well defined and understood operational mechanisms and information feedback from the manufacturing infrastructure. Casting the management of such information frameworks into well understood Operating System primitives can only aid in producing well organised and robust manufacturing management environments.

## 6. Conclusion

A framework was developed upon which future Flexible Materials Handling management systems can be based. The framework was derived from Operating System primatives to enable all Flexible Materials Handling systems to operate on the same governing concepts.

### References

[Bovet D. P and Cesati M., 2000] Daniel P. Bovet and Marco Cesati. *Understanding the Linux Kernel*. O'reilly, 2000.

[*Collett et* al., 2005] Toby, H. J Collett, Bruce, A. MacDonald and Brian, P. Gerkey. Player 2.0 Toward a Practical Robot Programming Framework. *Proceedings of the Australasian Conference on Robotics and Automation*, University of New South Wales, Syndey, Australia 2005.

[Gerkey B. P. and Mataric' M. J., 2004] Brian P. Gerkey, Maja J. Mataric'. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *International journal of Robotics Research*, 23(9):939-954, September 2004.

[MMHS-Metamorphic Materials Handling System - Final Report. 2003] *Metamorphic Materials Handling SystemFinal Report*, Doc No. IMS/ISC/11/6 – 3/1 Issue:1., 2003.