ELSEVIER

Applied Ergonomics 37 (2006) 91–110

APPLIED ERGONOMICS

www.elsevier.com/locate/apergo

Systems integration

C.E. Siemieniuch[a], M.A. Sinclair[b,*]

[a]Dept Systems Engineering, Loughborough University, UK
[b]Dept Human Sciences, Loughborough University, UK

Received 22 July 2004; accepted 14 June 2005

Abstract

The paper presents a view of systems integration, from an ergonomics/human factors perspective, emphasising the process of systems integration as is carried out by humans. The first section discusses some of the fundamental issues in systems integration, such as the significance of systems boundaries, systems lifecycle and systems entropy, issues arising from complexity, the implications of systems immortality, and so on. The next section outlines various generic processes for executing systems integration, to act as guides for practitioners. These address both the design of the system to be integrated and the preparation of the wider system in which the integration will occur. Then the next section outlines some of the human-specific issues that would need to be addressed in such processes; for example, indeterminacy and incompleteness, the prediction of human reliability, workload issues, extended situation awareness, and knowledge lifecycle management. For all of these, suggestions and further readings are proposed. Finally, the conclusions section reiterates in condensed form the major issues arising from the above.

1. Introduction

The aim of this paper is to present an ergonomics/human factors perspective on the world of systems integration, rather than a full, systems engineering view. This is in recognition of the worldwide spread of the 'Capability Maturity Model Integration' (CMMi), and the imminent spread of the 'Usability Maturity Model Integration' (UMMi). These two standards-based approaches provide frameworks and generic procedures for 'systems engineering, software engineering, integrated product and process development, and supplier sourcing', and are taking on the status of de facto standards for system integration capability. Accordingly, this paper is written within the context of these frameworks. Key sources for the CMMi, the UMMi and for Systems Engineering, at the time of writing, will be found at:

http://www.sei.cmu.edu/cmmi/
http://www.teraquest.com/CMMI/static/MainPage.html
http://www.processforusability.co.uk/Usability_test/html/umm.html
http://www.incose.org/

Guides on paper are numerous for the CMMi; see, for example Ahern et al. (2003). The UMMi has been published as ISO 18529 'Ergonomics of human-system interaction—Human-centred lifecycle process descriptions'.

This paper is in three parts. The first part ('Systems Integration Issues') begins by discussing the classical context of systems integration, partly as a reminder for readers and partly as a scene-setter for the second part. We then discuss a relatively new perspective, 'Complexity and the System of Systems', and a particular version of this, 'Immortal Systems'. The second part ('Systems Integration Processes') then discusses the process of performing systems integration, bearing in mind the context set out in the first part. It begins by outlining the CMMi, and the UMMi, both generic process models for systems integration, intended to help organisations

*Corresponding author. Tel.: +44 1509223002; fax: +44 1509223940.
E-mail address: m.a.sinclair@lboro.ac.uk (M.A. Sinclair).

0003-6870/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.apergo.2005.06.012

improve their systems design and integration processes. A number of ISO standards relevant for practitioners of human factors/ergonomics are listed, and the section ends with a generic, skeleton process for systems integration, that could be elaborated for a given integration scenario. The third part ('Systems Integration Problems') discusses a series of issues which are often encountered by practitioners, and makes a few suggestions for tackling them.

2. Systems integration issues

2.1. What is a 'system'?

We begin this section with this question because the definition ordains the nature of the discussion that follows. As is often the case there are as many definitions of a system as there are academics researching the area (see for example Emery and Trist, 1960; Machol, 1965; Churchman, 1968; Emery, 1969; Singleton, 1974). Of these, perhaps the best is the Singleton version, in that it is the most inclusive, and one that has been adopted by most organisations, including the CMMi (albeit in a more wordy version)—where a system is

- A set of interconnected entities
- The entities are dynamic in their behaviour (i.e. can exist in more than one state)
- The collection has a purpose or reason for existence

From the ergonomics/human factors perspective mentioned above, there is a corollary to this definition:

- The collection of entities includes people, processes and technology.

2.2. Key issues in systems thinking

We discuss here a number of the principal tenets in systems thinking, since these comprise some of the 'givens' that must be addressed in systems integration. A longer discussion of these, and more, will be found in Leonard (1984) and Skyttner (1996). Within each of the issues discussed below, connections are made to systems integration, the topic of this paper.

2.2.1. Boundaries and the environment

The critical issue here, with respect to systems integration, is where to place the boundary around the system under consideration, and how to treat the boundary.

Clearly, the closer in the boundary is placed, the lesser the number of parameters and variables that have to be considered explicitly but the more likely it is that significant behaviour will be omitted, or simplifying assumptions will be made, that turn out to be errors. Equally, the further out the boundary is moved, the more comprehensive will be the set of variables and parameters to be considered, and the more work in systems integration will be required; unfortunately, if the variables interact, the volume of work tends to increase logarithmically as the number of variables increases linearly.

The boundary itself can be considered impervious (in which case this is a closed system and no attention is needed outside the boundary), or a porous layer. The latter is by far the most common treatment (i.e. we are dealing with 'open' systems). Note that this is very important from a systems integration perspective; we can either treat what happens outside the boundary as 'noise', to which our system must be resilient, or we can treat it as an enclosing, structured system with which our system-of-interest must integrate.

Moray (1986) presented an illuminating example of some of these issues, and where to place the boundary. In Fig. 1, an operator/process system is shown, where A–F are sub-systems within the overall system. Note that the operator 'knows' about several functions, and the computer-controlled part 'knows' about others. And yet there are parts of this system (e.g. F in the diagram, which might be a sealed power supply which belongs to and is maintained by a public utility) that are known to neither. This is a fairly common scenario on many manufacturing shop floors. The question then, if computer control is to be upgraded by replacement, is where to place the boundary—exclude F? exclude A? Exclude the operator as well? How, then, should we treat the operator, F and A? As sources of noise, or of structured information?

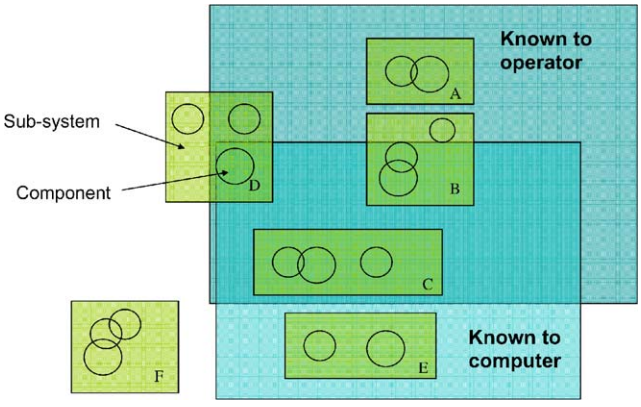

Fig. 1. An exemplar, originally developed by Moray (1986), which could be depicting a computer-controlled machine tool, with a number of human interfaces, and some unknown components as well. If the computer controller is to be upgraded, where should we place the system boundary when planning the upgrade?

2.2.2. Variation and delay

These issues are best encapsulated in Ashby's Law:

> Given a system with a regulatory process C, required to maintain a goal state G, but affected by a disturbance D (either from the process P or the environment N); the goal state G can only be maintained if the regulator C has sufficient variety and sufficient channel capacity to counter the variety in D. (Ashby, 1962)

Relating Ashby's Law to Fig. 1, the regulatory process C might correspond to the human operator; the goal state G would be that operator's goals for the process; the disturbance D might be a malfunction of subsystem B coupled with an unexpected reaction from the computer because of a knock-on effect of the malfunction in subsystem E.

For manufacturing systems, this means that the control system (hardware/software/liveware or a combination thereof) must be able to recognise and react to anything that could affect its safety and efficient operation and to do so sufficiently quickly. For automatic systems, this means that the designers must anticipate all likely disturbances over the expected lifetime of the system and create the means to counteract them, swiftly. The evident problem here is that it is very difficult to foretell the future; it constitutes one of the strongest reasons for including learning components of sufficient complexity (such as humans) within the control loop to encounter, understand, and react to the unexpected future. This, in turn, has a number of consequences for systems integration; if we now decide to include humans within the system as controllers and system adapters, and expect them to cope with a future that we may not predict, we must then include provision for learning to occur, together with support sub-systems for this learning.

Another issue on which to focus is that it is not just external disturbances which require attention; internal parts of the system will require attention, too. This implies that no system will remain in a situation of desired stability for long; indeed, the notion that the management of operational systems merely requires the detection of abnormal states followed by remedial action to return the system to 'normality' is false; for any complicated system, the likelihood is that 'normality' will require constant attention and management. For system integration, it follows that careful attention to control issues, and to the interfaces of the combined systems, is a crucial requirement.

2.2.3. 'Common mode' issues

As is well known in engineering circles, because of the interconnectedness of components within systems, it is possible for an event at one part of the system to affect other parts in unintended ways, through unanticipated channels. This may be because the components are closely packed (e.g. causing heat transfer problems), or because there are common inputs (e.g. all components draw from the same power source, or all human operators have been given the same, slightly out-of-date, description of the system), or a number of other causes. Unfortunately, common-mode issues tend to be subtle in their effects, and potentially disastrous interactions can be overlooked. Hence, there is a need to incorporate lessons from other similar systems, and precursor systems, to develop prototypes as early as possible and to carry out exhaustive risk analyses and integration and certification trials. This also applies to well-understood systems brought together to make a larger system. Systems are particularly susceptible to this at upgrade times, when the engineers upgrading the system are different from those which designed the system originally (for example, because of retirement), and do not have the same experiential knowledge as the original set of engineers.

2.2.4. The 'darkness principle'

This states that 'no system can be known completely'. While this is unprovable, it is likely true. This can be shown by a simple calculation; assume there are 10 interlinked components in a system, and further assume that each will be in one of two states: 'working' and 'failed'. Then, to check all possible states will require 512 checks. If we include just one class of intermittent faults, the number of checks required approximates 20,000. Rapidly, as system complexity increases, the resources required to execute these checks go beyond the capabilities of a design team.

Fortunately, it has been shown that there is a corollary to this principle; 'even though a system is never completely known, it can be managed effectively' (Lundgren-Cayrol, 2000); we are surrounded by systems which exhibit this behaviour. Nevertheless, from a systems integration perspective, this issue looms large. Systems integration implies that two existing systems will be merged; if each has its own 'basin of stability' (Ashby, 1962), then the merger may create chronic instability of the enlarged system (for example, as each of its constituent subsystems manipulates the joint parameters to achieve its prior stability). Accordingly, there is a requirement for extensive modelling of systems, and for transition periods during which the enlarged system can demonstrate its stability, and present at least some of its inherent surprises.

2.2.5. Federation and hierarchy

These lie at opposite ends of the spectrum of control. Hierarchical systems embody the notion of 'top-down' control; in principle, lower levels are told what to do and by when to do it, and in turn can deliver instructions to still lower levels. Necessarily, this is predicated on a

single hierarchy of control. At the other end of the spectrum is the federation; a number of separately controlled entities come together to deliver an outcome. Each of these has its own control structure, and common agreement is reached only by negotiation and persuasion. Immediately, this brings the nature of communications between them into focus and, in turn, this brings in the importance of trust and confidentiality (Siemieniuch and Sinclair, 2000). In short, without trust in other agents, and belief in their integrity and ability to keep confidential information confidential, there will be little or no communication, however good the communications infrastructure might be. And communications, at the four levels of transactions, operations management, policies and strategic issues, are required to secure the co-operative agreements necessary for a federated system to work well.

Arguably, these issues apply to any systems integration project; in developing a system for integration into another system, there will be both hierarchies and federations to be considered—federations are a characteristic of supply chains. Equally, integrating that system over there designed by those people over there into our system here will involve considerations of federation. In addition, from the paragraph above on 'darkness', this will be especially important if that system over there brings unknown complications with it when it is integrated into our system. If this should happen, then it is the quality of the relationships that have been created between people that will matter most in reaching a quick, satisfactory solution, not the nature of the penalty clauses in the contract, whatever their longer-term effect might be.

2.2.6. System entropy

Following the Second Law of Thermodynamics (Atkins, 1986), all the components of systems, left to themselves, will decay. Given that this 'decay' is happening within a non-stationary environment, likely to change in a discontinuous manner, it follows that maintenance effort and upgrades will be essential to counteract the growth in entropy. In other words, continuous management effort is likely to be required to maintain normality, as argued earlier. And this will only work if suitable processes and infrastructures are in place—making another issue for consideration in systems integration.

Legacy systems are a particular example of systems entropy in action. This refers to existing systems, often well-used and embedded deep in the organisation's core processes, which over the years have been altered, adapted, upgraded, and persuaded to do things they were not originally designed to do. In principle, all such changes will be documented and all the accumulated, associated knowledge will be carried forward; in practice, this is not often the case. More commonly, key documents are lost or misplaced; others are on media which can no longer be read; not all the documents were completed in the first place; and those people who did know how the system worked and who had the tacit knowledge that comes from operation may well have left or retired. In short, the legacy system may work well, but nobody will be sure why, and even fewer people will be willing to tamper with it.

The preceding paragraphs all presumed the availability of information and knowledge about the systems to be integrated; with legacy systems this typically is faulty, and in turn this implies that systems integration will be difficult and error-prone. It is a brave systems integrator who goes into this situation; blame and denigration of your efforts are highly likely, and the consequences of disruptions may be catastrophic. A recent example happened in the UK air traffic control system for Heathrow and other airports around London; a single legacy-system computer application stopped working while upgrading was being completed, causing a collapse of the new air traffic control system. The results were benign in that the controllers were able to use an older system and ensured that no airplanes crashed, but many were grounded, thousands of passengers had difficulties with their travel arrangements, and the financial costs have yet to be quantified at the time of writing.

2.2.7. Effects of complexity

Many authors make a distinction between 'simple' and 'complex' and/or 'complicated' and 'complex'—see Cilliers (1998) for a revealing discussion on this aspect. For the purposes of this paper the authors distinguish between 'complicated', and 'complex'. 'Complicated' systems may be difficult to understand, but in essence behave the way they were designed to behave, and can be decomposed to understand their behaviour. 'Complex' systems, on the other hand, are able to evolve their behaviour, and so perform in ways that the system designers did not necessarily intend—complexity hence arises as a result of non-linear interactions between elements in the system often producing what is referred to as 'emergent properties', which are not likely to be demonstrated under decomposition. Characteristics of complex systems include the following:

- Many agents, of different kinds usually with some organic components
- Some degree of behavioural autonomy and intelligence for agents (e.g. some are human, and can think)
- Multiple steady states for agents
- Lack of equilibrium in the system
- Non-linear interactions between agents in an environment
- Lots of connections between agents
- Agents communicate in parallel

- Effects of operating in an evolving environment
- Effects of evolving agents
- Interactions between different goals within an agent
- Interactions between agents with different goals
- Language/culture differences

Given that a sufficient subset of these characteristics are present, then emergent behaviour can be expected, but from the arguments above the type of behaviour is likely to be unexpected. In practical terms, one outcome of this thinking is the de Meyer et al. classification of projects (deMeyer, Loch et al., 2002)

- *Variation projects*. Gantt charts and PERT charts are appropriate to manage these; the project management aim is to maintain 'normality'.
- *Projects dealing with foreseen uncertainty*. For example, it is known that competitors are going to bring out a new product and a reaction will be required; however, on the basis of market knowledge, one could guess likely attributes. Projects of this class require risk management techniques and decision trees, to enable a quick adjustment to whatever happens.
- *Projects dealing with unforeseen uncertainty* (i.e. there is no Plan B). The project team for these projects need to be flexible orchestrators, networkers, and ambassadors. Trust is critical, and the project team must scan horizons continuously, and enter into flexible contracts, whenever possible.
- *Projects in a chaotic environment* (usually due to complexity of links into the environment, extending internal complexity). For these, the project team must be entrepreneurs and knowledge managers. Success in these projects depends critically on having long-term relationships which go beyond individual projects. Project relationships must be based on trust and handshakes, rather than on contracts, with a continuous, ruthless concentration on go/no-go decisions. Typically, plans exist only to the next decision gate. Under these circumstances, the most feasible way forward is to adopt a 'rapid prototyping' approach, to find problems and to acquire knowledge as early as possible, at minimum resource cost. It is also obvious that conflict management is a necessary skill for all participants.

It might be asked, 'what about systems in which only part exhibits complex behaviour?' The usual answer is to see the parts in terms of the de Meyer et al. classification above, and to engineer them accordingly.

From a systems integration perspective, complexity has a profound impact. The more that complexity is an issue, the more the integration process must be flexible and empiric, with design and integration decisions delayed as late as possible. It is also obvious that the integration project must look further into the operational environment than just the immediate interfaces to the rest of the system (including the environment) to obtain the best understanding of what the future might entail. It is also obvious that a very modular system architecture would be ideal, to enable re-engineering of the system to be accomplished as economically as possible, both in the light of emergent behaviour and in recognition of the system's possible immortality, as discussed in the next section. There is a strong implication that rapid prototyping, perhaps by simulation techniques, represents the best way forwards; not just for design of the system, but incorporated into the system as delivered and integrated, to give the personnel involved the best chance possible of finding emergent behaviour before it actually happens. Finally, for the system operators/controllers, Rasmussen's dictum (Rasmussen and Goodstein, 1986) that they should 'be seen as the designers' representatives on-site' is relevant; if emergent behaviour is likely, then the operators who will have to confront this must understand not only how their systems work, but why they work that way. Without this knowledge, it is likely that their efforts, based, as they will be on what logical analyses they can perform, will be insufficient, and possibly disastrous.

2.2.8. System lifecycle, and the 'immortal system'

Fig. 2 illustrates an example of a system lifecycle, from the domain of defence, but characteristic of the models being developed in many other domains where a 'service' approach is being adopted.

Fig. 2 neatly encapsulates one of the big difficulties of systems integration; at the 'main gate' decision activity (where the necessary resources are committed) the assembled team of stakeholders, comprising suppliers and customers, are expected to come to an agreement about the user requirements specification, in the knowledge that the lifespan of the system might be decades, operating in a changing environment. In some cases, this might be a century or more; an example is the USAF B-52 bomber. This was originally designed to deliver nuclear weapons over very long distances across unfriendly territory. It is now used to provide front-line support in localised wars; a marked change since the airplane was first envisaged; there has been at least one 'revolution in military affairs' since then (O'Hanlon, 2000). Clearly, from a systems integration perspective, this places critical pressure on the prescience of the team, and on the skills of the engineers in devising an appropriate product breakdown structure (PBS) to accommodate these requirements, knowing the requirements may turn out to have been inappropriate. In principle, the architecture of the PBS should be highly modular, to enable the planned and unplanned upgrades to happen as efficiently as possible, as discussed above. As a corollary to this, Hippel (1990) has shown that an

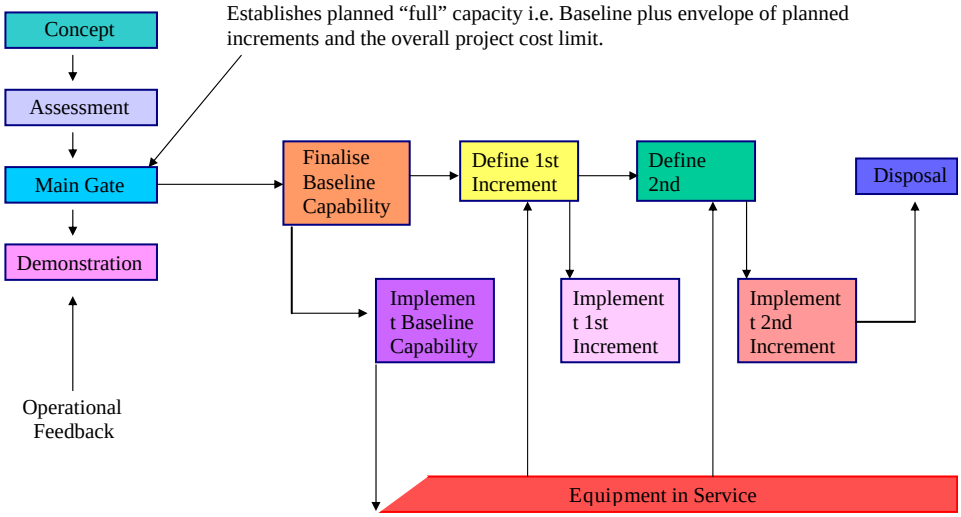

Fig. 2. Example of a system lifecycle, from the UK Ministry of Defence 'Smart Procurement' initiative. The activities to the left constitute the development of a requirements specification; those progressively to the right constitute the lifecycle of a given system from then on. Only two upgrades are envisaged in this diagram.

inappropriate PBS has a big effect on time to completion. From a more recent perspective, it is tempting to think that this might be due to induced complexity in the design process.

The other significant issue in Fig. 2 is the effect of system immortality. The immediate consequence is that nothing will remain the same:

- The designers and human operators will not be around for more than 40 years, and it is likely that most will have moved on within 20 years. This has two effects; firstly access to the tacit knowledge used for the design and operation of the system may be shut off, unless there is a support process to obviate this, and secondly it places a premium on good software documentation—that it should be complete, and that it should still be available. As an example of the latter, it is still possible to read the Domesday Book written in A.D. 1068 that listed property ownership in the UK, whereas a similar BBC exercise in creating a more modern Domesday Book on life in the UK in the 1980s is essentially unreadable, because the laser disks and associated hardware on which it was recorded at the time are no longer supported.
- The supply chain that supports the system may change markedly; some companies will cease operations, others will merge, perhaps with competitors. This may have a severe impact on the system's support. Even if the supply chain is still in existence, the production processes may be dismantled, or re-commissioned for other purposes.
- Requirements for the system will evolve as the system progresses through its lifecycle. Likewise the architecture of the product may change, as components are progressively replaced.

From a systems integration perspective, the concept of system immortality emphasises several things:

- The importance of knowledge management, to address the inevitable personnel and organisational changes
- Standardisation in design, to alleviate the problems of disintegration of the supply chain
- Modular design, to economise on the resource costs of the upgrades to come for the system
- The use of a model-driven design approach, to speed the design process, to make best use of commercial off-the-shelf (COTS) equipment, and to reduce the effort of the upgrading that will happen
- The inclusion of 'proven-best-in-class' components, because they are likely to be more available, and likely to perform their roles for longer before upgrading. Note that the components need not be 'best-in-class'; there are many integration issues stemming from different interpretations of common standards, so it is sometimes better not to be first in using the best components.
- The utilisation of IT-based components rather than hard-engineered components, because the upgrading effort and logistics are likely to be less, and because there is likely to be further integration of systems as time passes and the operational environment changes.
- The importance of human-centred design, to reduce the operational requirement for specialists and thereby enlarge the selection pool, to make the training for operators and maintainers more comprehensible and efficient, and to make it easier to deliver the 'why' training as well as the 'how' training.

3. Systems integration processes

It is now time to consider how systems integration can take place, given the considerations above. We have adopted a 'process' perspective for this, based on the premise that systems integration is achieved by the execution of a series of appropriate activities. Given this, it is incumbent to utilise mature, well-designed processes. From this perspective, the two key approaches are the CMMi and the UMMi.

3.1. The 'capability maturity model integration' (CMMi)

This process was developed by the Software Engineering Institute (SEI) at Carnegie-Mellon University, for 'Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing'. The main purpose of this is to improve the functionality, quality and efficiency of organisational systems. This comes in two versions: 'continuous' (V1.1) and 'staged' (V1.2). The concept of capability maturity stems from work by Crosby in the 1970s (Crosby, 1979), and Fig. 3 represents his ideas, in the context of quality. The SEI then extended these ideas for the domain of software engineering, and subsequently for the whole domain of systems engineering.

Two versions of the CMMi exist, to cater for different requirements in organisations. One is the 'continuous' version, which presumes there will be continual improvements to relevant processes, running in parallel. This is a suitable approach for companies embarking on the maturity voyage, up to stage 3. From there on, the 'staged' version is considered appropriate, so that improvement exercises are undertaken in step, reflecting the more integrated nature of the firm's processes at these later stages. The six stages ('levels' in the CMMi) are: 0—Incomplete (perhaps better called 'inept'), 1—Performed, 2—Managed, 3—Defined, 4—Quantitatively managed, and 5—Optimising. For each of these, goals and sub-goals are identified, together with processes and sub-processes that will enable the goals to be met. Progression from one level to the next is addressed.

3.2. The 'Usability maturity model integration' (UMMi)

This was generated by the Telematics Applications Project IE 2016 (INUSE) within the Framework Programmes within the European Union, aimed specifically at processes for human-system integration by means of human-centred design. A version of the UMMi is available at:

http://www.processforusability.co.uk/Usability_test/html/umm.html

Analogous to the CMMi, this is intended to provide organisations with the right, quality-assured processes for delivery of human-systems integration. Fig. 4, below, outlines what UMMi offers. Good introductions to the approach will be found in Sherwood-Jones (2003) and Sherwood-Jones, Earthy et al. (2003); these papers outline the standards which embed the UMMi, and discuss their inter-relationships, including those that

Measurement Categories	Stage 1: Uncertainty	Stage 2: Awakening	Stage 3: Enlightenment	Stage 4: Wisdom	Stage 5: Certainty
Management understanding and attitude	Fails to see quality as a management tool	Supports quality-management in theory but is unwilling to provide the necessary money or time	Learns about quality management and becomes supportive	Participates personally in quality activities	Regards quality management as essential to the company's success
Quality organisation status	Quality activities are limited to the manufacturing or engineering department and are largely appraisal and sorting	A strong quality leader is appointed, but quality activities are focused on sorting, & limited to engineering & manufacturing	Quality department reports to top management, and its leader is active in company management	Quality manager is an officer of the company. Prevention activities have become important	Quality manager is on the board of directors. Prevention is the main quality activity.
Problem handling	Problems are fought as they occur and are seldom fully resolved; "fire-fighting" dominates	Teams established to attack major problems, but approach remains short term	Problems are resolved in an orderly fashion, and corrective action is a regular event	Problems are identified early in their development	Except in the most unusual cases, problems are prevented
Cost of quality as percentage of sales	Reported: unknown Actual: 20	Reported: 5% Actual: 18%	Reported: 8% Actual: 12%	Reported: 6.5% Actual: 8%	Reported: 2.5% Actual: 2.5%
Quality improvement actions	No organised activities	Activities are motivational and short term	Implements the 14-step program with full understanding	Continues the 14-step program and starts Make Certain	Quality improvement is a regular and continuing activity
Summation of company quality posture	"We don't know why we have quality problems"	"Must we always have quality problems?"	"Because of management commitment and quality improvement programs, we are identifying and solving our quality problems"	"We routinely prevent defects from occurring"	"We know why we don't have quality problems"

Fig. 3. A 'Capability Maturity' matrix, addressing quality, from Crosby (1979). Note this demonstrates only five stages of maturity, whereas the CMMI defines six.

HCD 1	HCD 2	HCD 3	HCD 4	HCD 5	HCD 6	HCD 7
Ensure HCD content in systems strategy	**Plan and manage the HCD process**	**Specify stakeholder and organisational requirements**	**Understand and specify the context of use**	**Produce design solutions**	**Evaluate designs against 'requirements**	**Introduce and operate the system***
represent stakeholders	consult stakeholders	clarify system goals	identify user's tasks	allocate functions	specify context of evaluation	manage change
collect market intelligence	plan user involvement	analyse stakeholders	identify user attributes	produce task model	evaluate for requirements	determine impact
define and plan system strategy	select user-centred methods	assess H&S risks	identify organisational environment	explore system design	evaluate to improve design	customisation and local design
collect market feedback	ensure a user-centred approach	define system	identify technical environment	develop design solutions	evaluate against system requirements	deliver user training
analyse user trends	plan HCD activities	generate requirements	identify physical environment	specify system and use	evaluate against required practice	support users
	manage HC activities	set quality-in-use objectives		develop prototype	evaluate in use	conformance to ergonomics legislation
	champion HC approach			develop user training		
	support HCD			develop user support		

Fig. 4. Major processes in the UMMi, providing the capability for high-quality human-system integration. From Sherwood-Jones (2003).

could apply to the profession of ergonomics/human factors itself. They are:

- ISO 9126:2000 Software product quality—quality model.
- ISO 12207:1995 Software process—software process lifecycles. This includes a usability process.
- ISO 13407:1999 Human-centred design processes for interactive systems. This is written from a project-management perspective, and defines basic human-centred design principles and activities.
- ISO/IEC 15288:2002 Systems engineering—system lifecycle processes. This sets out the processes over the lifetime of the system which systems integration should deliver, including management of these.
- ISO TR 15504:1998-2002 Software process assessment—a reference model for processes and process descriptions. This is a meta-standard, describing what the other standards must deliver.
- ISO PAS 18152: 2003 A specification for the process of assessment of human-system issues. This comprises the UMMi, shown in Fig. 4, representing a sequence of design stages with sub-tasks within them.
- ISO TR 18529:2000 Ergonomics of human-system interaction—human-centred lifecycle process descriptions. This is illuminated by the 'Human-System' model from the ISO PAS 18152. In effect, this specifies what other engineers can expect from ergonomists/human factors specialists, and how they will contribute and hence relate to other system processes. It contains a number of processes, for delivery of results to the processes outlined in ISO15288. There are 4 of these:
  - HS1 lifecycle involvement. The processes in this are related to anticipating and tracking the particular needs of each stage in the lifecycle.
  - HS2 integration of human factors. The processes within this are concerned with strategy, procurement and planning, and ensure that human-system issues are addressed by the appropriate stakeholders. It reduces lifecycle costs by ensuring that 'design for people' is used within the organisation.
  - HS3 human-centred design. These processes are concerned with design for people who are using the system of interest in the context of use, addressing requirements, design and evaluation. It is a development of ISO TR 18529.
  - HS4 human resources. The processes deal with the provision of the correct number of competent staff for the system of interest; selection, training, harmonising organisational development, and technological change—it is about implementation, rather than design.

3.3. An exemplar framework for systems integration

We offer below a framework for systems integration which could be tailored to fit most scenarios. The framework has been included to show how the CMMi and the UMMi could be utilised, and also because of its emphasis on the human and managerial aspects of systems integration, not just the technological issues.

Figs. 5–14 comprise a generic capability development and deployment process (CDDP) model that was developed for use within the aerospace industry for governing the selection and insertion of new capability (people, process and/or technology) into an organisation. Fig. 5 provides an overview of the CDDP.

The assumptions underpinning this framework are that (a) a 'stage-gate' approach is appropriate for most systems integration projects; (b) the framework is at a sufficiently higher level that almost any design approach could be fitted within this process—e.g. rapid prototyping (Chua, Leong et al., 2003), or the VEE model (Forsberg and Mooz, 1992, 1995), or a waterfall approach (McConnell, 1996); and (c) for 'small' projects, it is feasible to omit portions of the framework as the users wish. The scope of the process is from initial vision to delivery of an integrated system; it does not include the rest of the system lifecycle.

At bottom is the stage-gate approach adopted by the organisation. The Integrated Business Plan maps out the series of projects expected to be performed in the current and future budget periods. A number of individual projects are shown, divided into eight stages. The projects are managed as a portfolio, having been incorporated in the IBP. In the diagrams that follow, time runs from left to right.

Figs. 7–14 will provide extrapolations or overviews of the activities required in each of the phases or stage gates illustrated in Fig. 5 above. In most cases, a phase review is held after each stage gate which the project must pass before it is allowed to continue. Fig. 6 provides an overview of the sorts of activities required both pre-and post reviews.

Each of the stage gates shown in the figures below comprise a set of management activities and a set of technical activities, interlinked by a communications 'bus'. Note that in 'execute system development' there are activities not usually included in academic discussions of system integration. As a reminder for the reader the eight stages are:

- Manage and approve Vision Phase
- Manage and execute Feasibility Study
- Manage and execute Detailed Planning for System Project
- Establish Project and utilise Resources

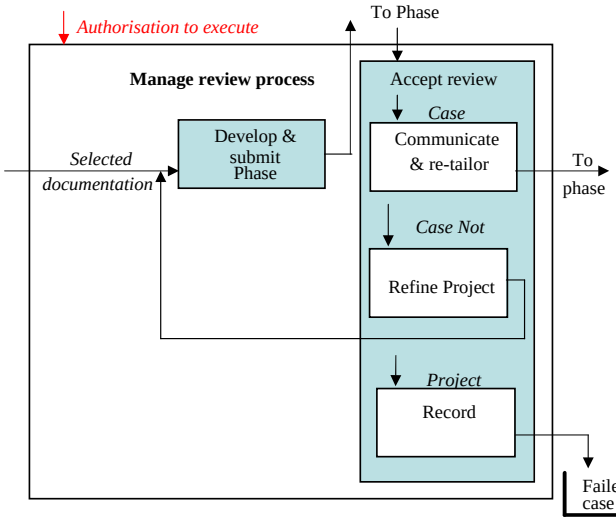

Fig. 6. Review activities, common to all stages, and follows activities of each stage. Note that the 'bucket' at right is a store for failed cases,

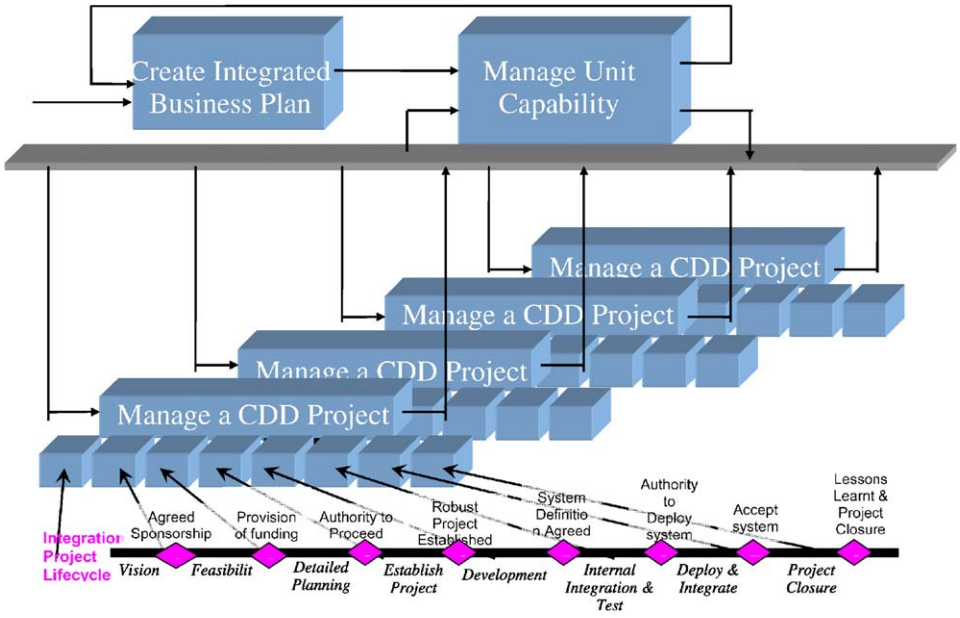

Fig. 5. Overview of CDDP, showing both individual system integration projects and the overall portfolio of these.

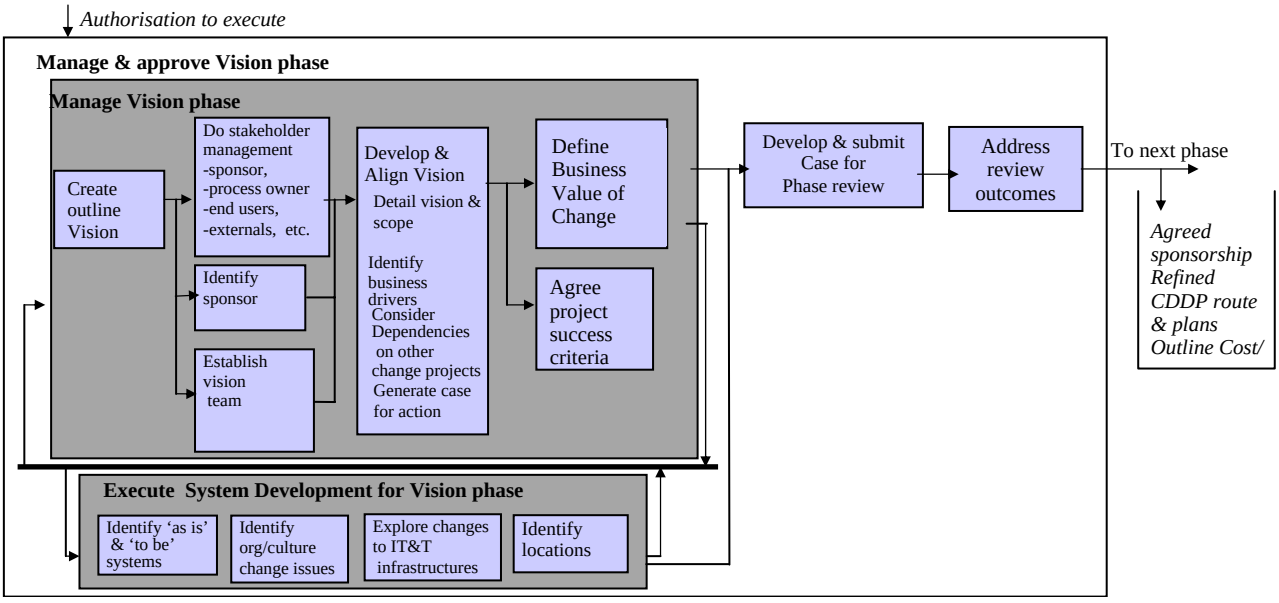

Fig. 7. Vision phase, outlining the integrated view of the project. The top half of the diagram deals with managerial issues; the bottom half with the vision content. These are joined by an information bus, across the middle of the diagram.

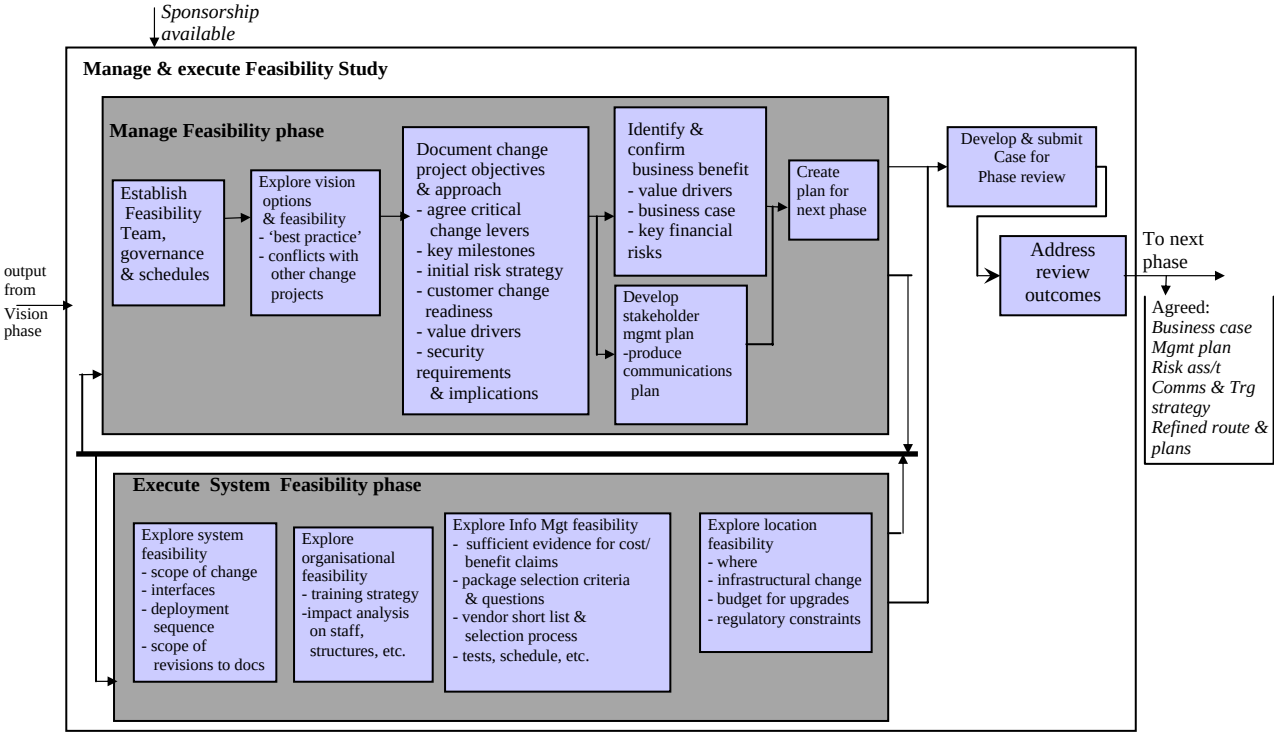

Fig. 8. Feasibility Study. Note continuity from activities in Fig. 7 to activities in Fig. 8. These continuities persist, as parallel strands of activity, until the process is complete. Initial activities for system integration outside the system-under-development are clearly evident at this early stage. The implication is that 'living documents' should be created, perhaps from templates, for each of these strands.

- Manage and execute Development Phase of System Project
- Manage and execute internal Integration and Test phase of System solution
- Manage and execute Deployment phase—system integration
- Manage and Execute Closure of System Project

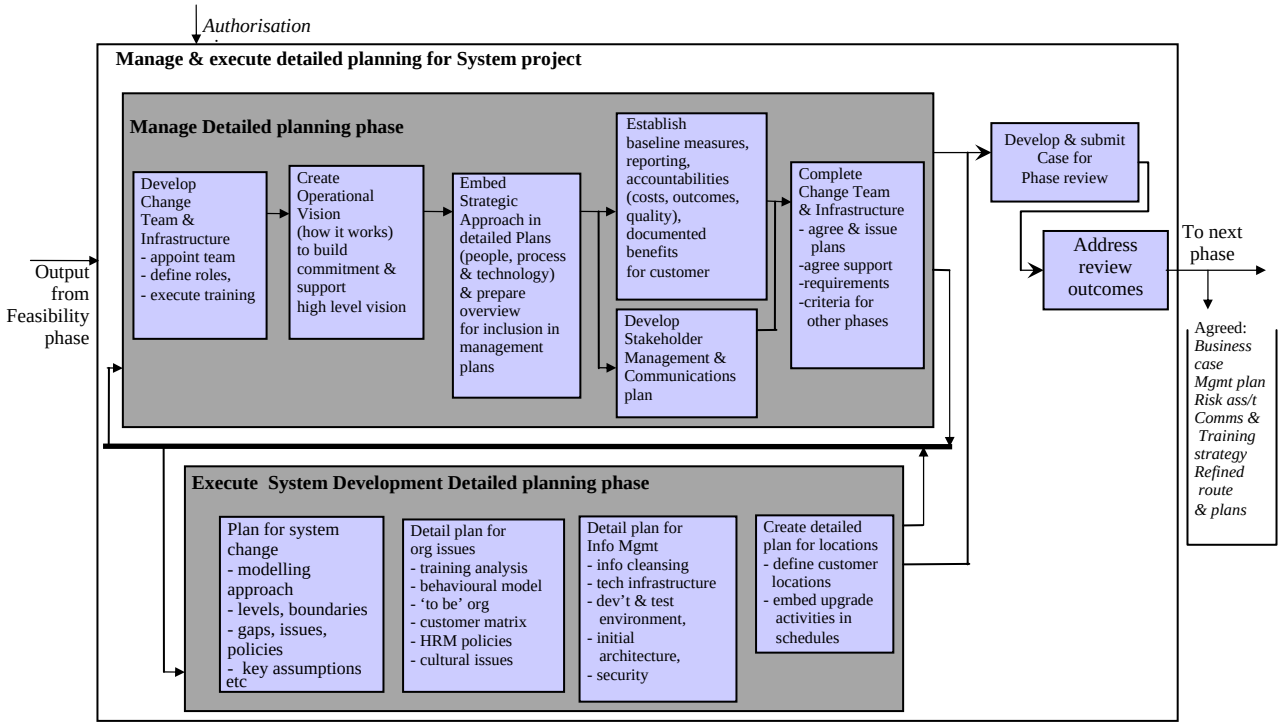

Fig. 9. Detailed Planning. Note that considerable effort is going into these initial activities, with the intention of minimising surprises and reworking later in the process. Organisational learning is an implicit part of all these activities.

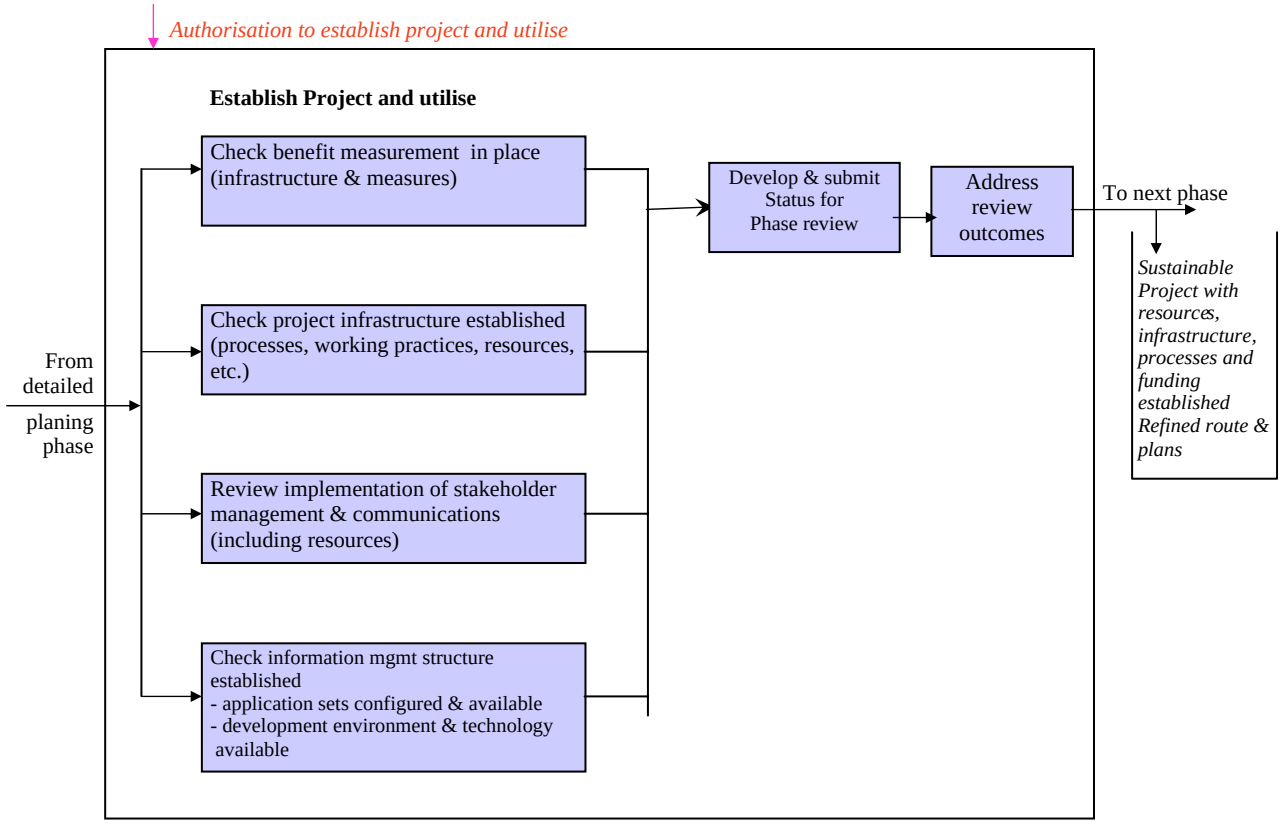

Fig. 10. Establish Project. From this point resources, especially money and people, are committed top the project and development work begins- a significant decision point in the project lifecycle.

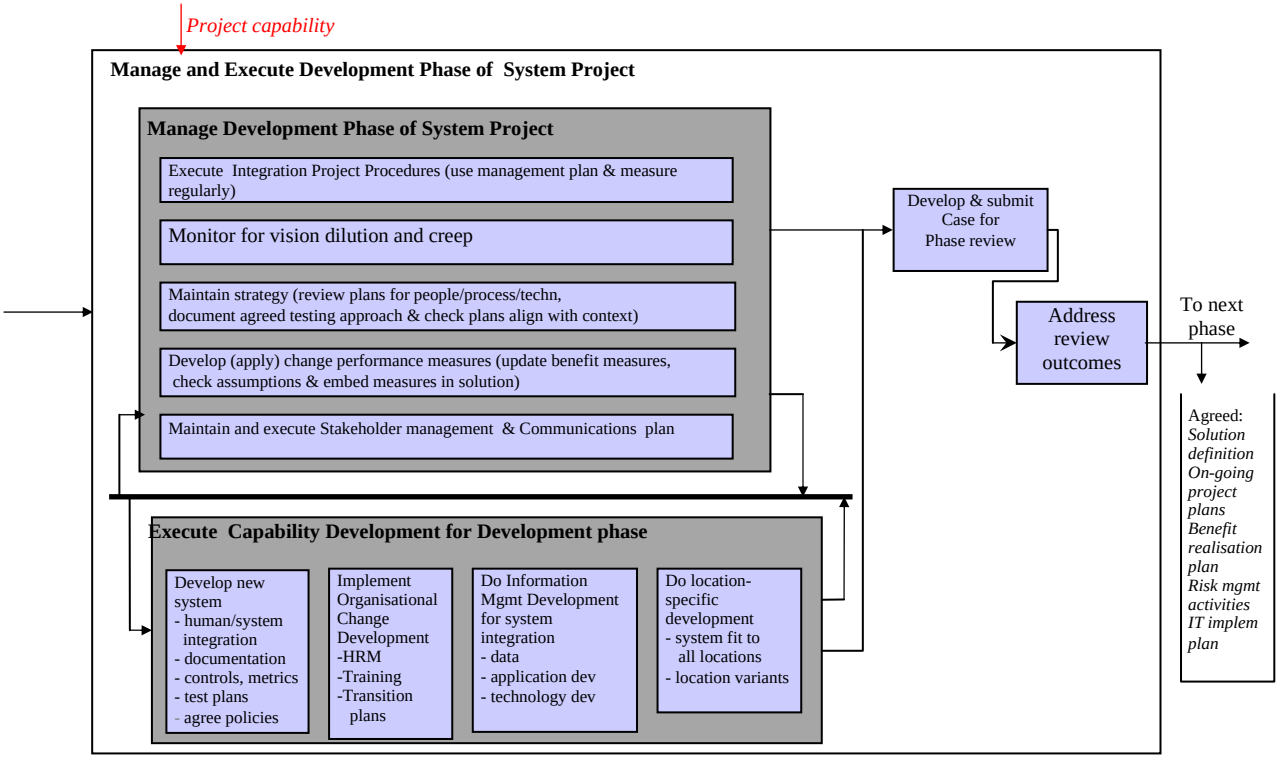

Fig. 11. Development Phase. In previous phases it is expected that considerable communication between the activities will have taken place. From here onwards, this will be much more intensive, and the 'living documents' will become more and more important, both as an audit trail and for knowledge capture and management. It is expected that considerable effort will be going into organisational learning, although this activity is not explicit in the diagram. This phase creates or re-uses system components (people, process and technology), or acquires them from the supply chains, using rapid prototyping or other approaches.

The particular approach as stated above has some advantages, and naturally enough disadvantages. The advantages are as follows:

- The CDDP approach does specifically address the human and organisational aspects of systems integration, as well as the usual technological aspects.
- It is designed for large-scale projects, and can be easily scaled down for small projects (scaling-down is safer and more easy than scaling-up).
- It is prescriptive, but with explicit recognition of the need for tailoring to meet the needs of individual, specific, integration projects.
- It covers the gamut from initial idea to final closure of the project.
- It includes consideration of the portfolio of integration projects that an organisation might find itself involved in managing.

The disadvantages are as follows:

- The diagrams as shown are very high level, and considerable development of them would be required for a practical exercise.
- The approach has a resemblance to the waterfall model of project management, in the form shown above. It can be adapted to a spiral model, but further work would be required by the practitioner to develop this.
- The diagrams do not show the 'living documents' approach that would be necessary for this framework to work This set of documents provide both knowledge capture and an audit trail for the project, and either this set, or an analogous, in-house set, would be necessary if this approach was to be adopted.

However, even armed with all the knowledge outlined above, the practitioner will still find problems likely to cause a distinctly furrowed brow. We discuss briefly some of these, pointing to possible approaches that might be useful in a given situation. In addition, useful advice will be found in the 'Handbook of Human Systems Integration' (Booher, 2003) which is filled with a wealth of useful information and examples of human systems integration. It includes information on simulation-based acquisition, a user-centred systems engineering framework, integrating training into the design and operation of complex systems, system safety principles, and cost-benefit analysis for this domain. Other useful texts include Chapanis (1996), Stolovitch and Keeps (1999) and Rhodes (2000).

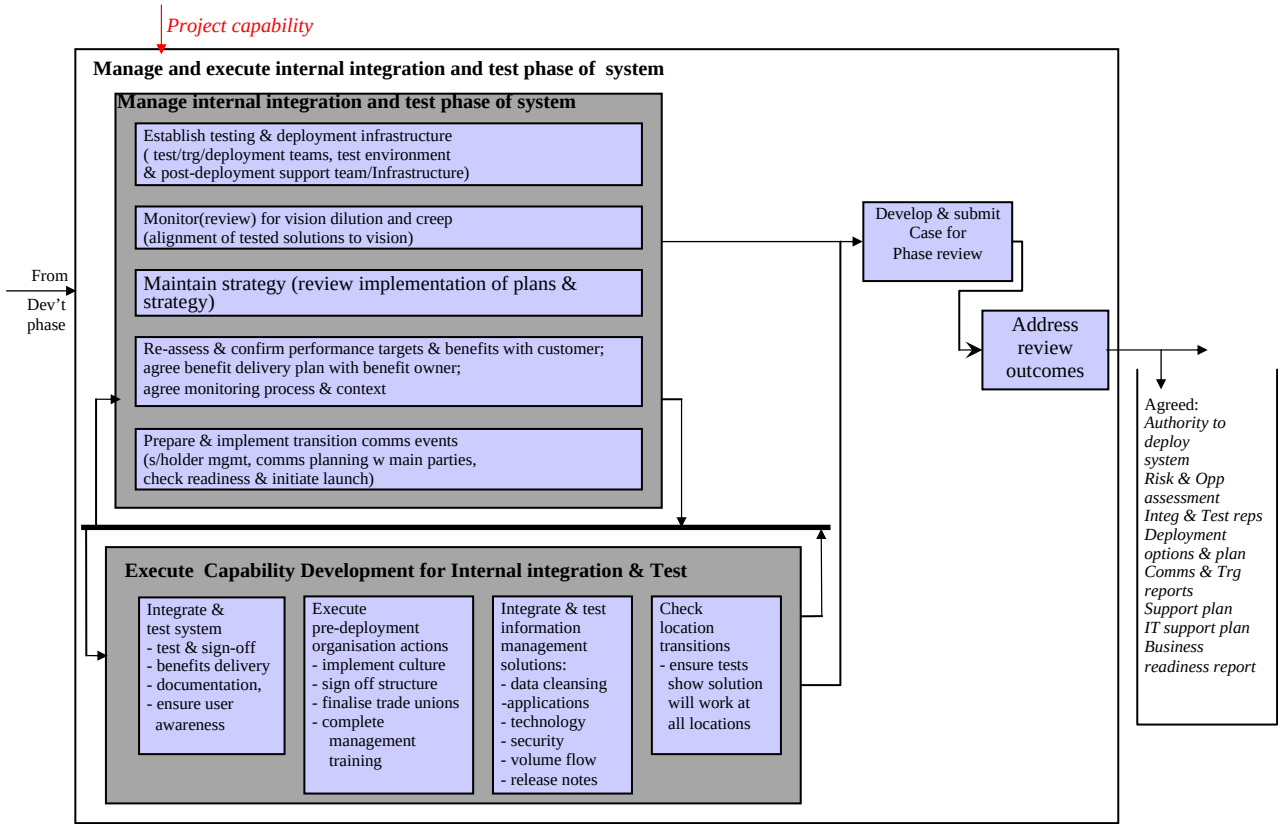

Fig. 12. Integration and Test Phase. People, process and technology are checked for integration and inter-operability to create the whole system

3.3.1. *The provision of autopoiesis*

Despite the rise in importance of genetic algorithms (with their inbuilt powers of adaptation and evolution) and other technological advances, it is likely that for the foreseeable future it will be humans who will be responsible for the initiation and execution of system change. In the context of systems integration, this likely be so for two reasons; legal issues, and because of the need to control emergent behaviour. The latter demands some degree of scanning of the other systems with which the system-under-review interacts, and a degree of tacit knowledge of likely problems that can only be brought into a given system from outside. Perrow, in his book 'Normal Accidents' (Perrow, 1999), indirectly makes this case. If it is accepted that autopoiesis will be the responsibility of humans, then an immediate consequence is that there needs to be a culture of organisational learning, and sound support (training, tools, processes, job rotations, etc.) for those charged with the responsibilities for this.

The other issue that may also have some bearing in the future is the problem of computer security in general, and viruses in particular, some of which may act as an alternative nexus of control for the system. In integrated systems this is of critical importance that needs to be addressed; unfortunately, your authors are unable to help here, as these issues are outside of their competence.

3.3.2. *Incompleteness, and the minimum critical specification*

This is a well-established principle within the socio-technical world-view (e.g. Eason, 1988; Ehn, 1988; Eason and Harker, 1991; Eason, Harker et al., 1997). This is predicated on the notion that the system stakeholders, and the end-users in particular, will necessarily adapt the system as designed to fit better with the dynamic reality in which they find themselves. Accordingly, it would be wise only to specify that which has to be specified, and leave the other aspects open to adaptation. While this is a sound notion in principle, it is more difficult to achieve; for example, where would design authority reside? Who would be responsible in the case of accidents? This approach is perhaps best brought to fruition by ensuring very close collaboration with the stakeholders, building in local control loops such that decisions are made close to where the issues arise, and creating an organisational culture where leadership is prevalent, evident, coherent and cohesive. Fortunately, a new methodology has been suggested (Herrmann, Hoffmann et al., 2004) which might offer progress here, and for the next topic too.

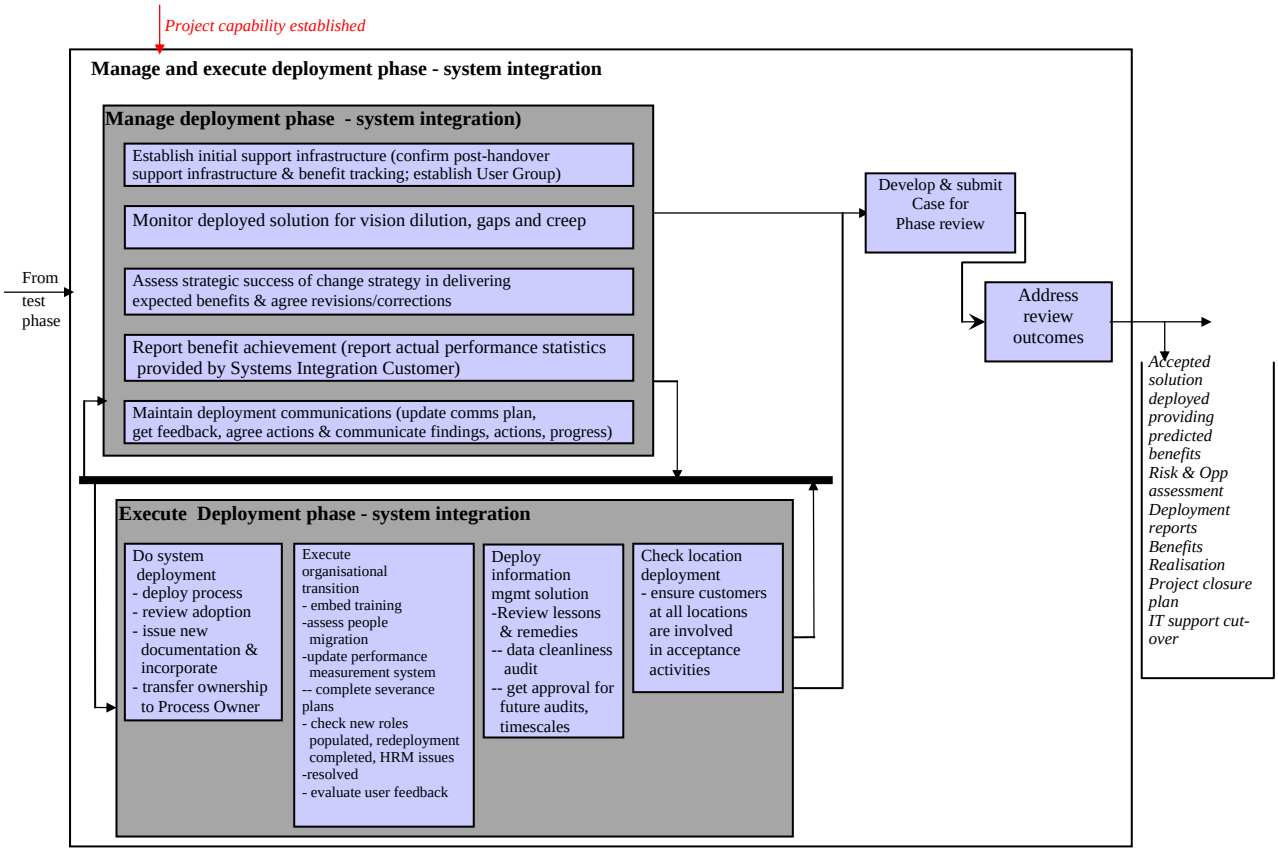

Fig. 13. Deployment Phase. The system is now integrated into its operational environment, assumed to be some larger system. This may be accomplished by various approaches; 'big bang', parallel operation, or incremental, for example.

3.3.3. *Indeterminacy*

This reflects one aspect of human behaviour; what Reason has called 'human as hero' as opposed to 'human as error source' (Reason, 2001). From the 'human as hero' perspective, it is accepted that a significant role for humans in the system is as system saviours, the safety shield of last resort. This reflects the notion that systems design will always be incomplete, usually due to complexity, 'darkness', limitations on the resources that can be allocated to design (chiefly time and money), and the inability to predict accurately the scope of future uses of the system. It is therefore left to the operators of the system to make up for control omissions and problems lurking in the system; examples of this are the Three Mile Island incident (eventually, humans brought this system under control), Apollo 13, and similar. In effect, humans are used as the environment/system control interface. For this role, it is necessary that the operators are made fully aware of systems behaviour; as Rasmussen said, many years ago, "systems operators should be regarded as the on-site representatives of the designers" (Rasmussen, 1986), with appropriate interfaces to allow them to understand and exercise control, for example via ecological interfaces (Vicente and Rasmussen, 1988; Vicente, 1990; Vicente, Burns et al., 1997; Vicente, 1998). But they need more than this; both Ashby's law and the complexity argument indicate that there will be unexpected uses for the systems, unexpected behaviour of the systems in certain circumstances, and 'unknown' problems to solve. Furthermore, 'normal' operation of the systems will allow skills to become rusty over time (especially for the rare problems). All these problems provide a clear need for very good simulation tools. Since most systems projects from now onwards are likely to include system models for testing and evaluating the system, it is not necessarily a large step to include simulation facilities. But, by definition, the simulations will be incomplete, so there will need to be editing support to enable the simulation tool to be upgraded to reflect current understanding. In turn, this provides another case for simulation, it is a means of capturing a particular class of formal knowledge as organisational learning, without which the organisation could find itself in legislative and competitive trouble.

3.3.4. *Human reliability assessment*

In a nutshell, this is based on the 'human as error source' perspective. There is a plethora of techniques available (e.g. HEART (Williams, 1985), CREAM

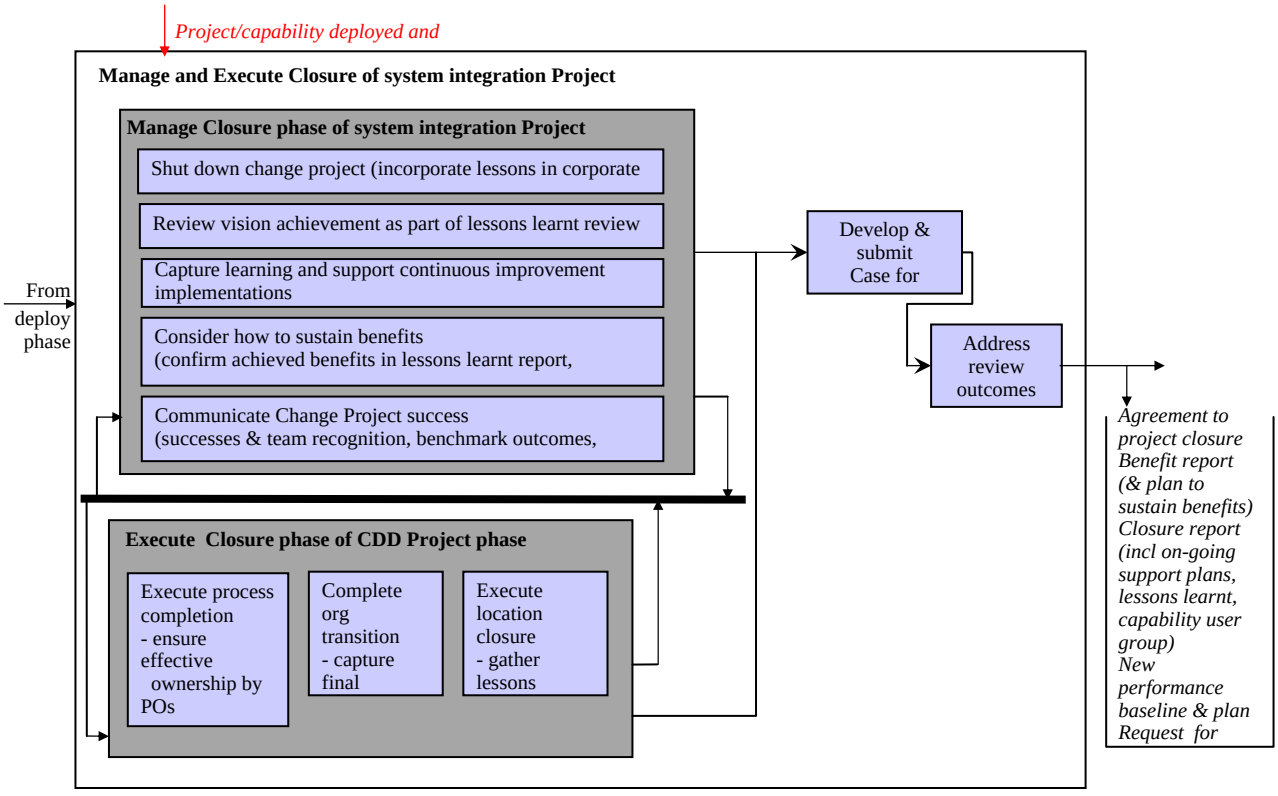

Fig. 14. Project Closure. System integration is concluded, and organisational learning is completed and embedded within the organisation. Note that this includes tacit knowledge.

(Hollnagel, 1998), THERP (Swain and Guttman, 1980), SLIM-MAUD (Embrey, Humphreys et al., 1984), THEA (Pocock, Harrison et al., 2001)) and see also Kirwan (1994), but there are a number of methodological problems in this area. One critical review (Dougherty, 1990) adumbrated:

- Less-than-adequate data
- Less-than-adequate agreement in use of expert-judgement methods, both agreement between judges and accuracy of predictions
- Less-than-adequate calibration of simulator data, because simulators are not calibrated either
- Less-than-adequate proof of accuracy in HRAs, particularly for non-routine tasks
- Less-than-adequate psychological realism in some HRA approaches
- Less-than-adequate treatment of some important performance shaping factors (PSFs) such as managerial methods, cultural differences, and interpersonal variability

To which could be added:

- There is no recognised process for HRA within the design cycle; the missing HRA process should include a toolbox of methods for different phases of the cycle (e.g. error identification at early stages of design, followed by more and more rigorous and quantitative methods as the design matures, engineers need the numbers, and validation and verification approaches. The process should also address training and management of HRA knowledge, benchmarking, and management of the process itself.
- None of the numerical methods is particularly accurate; "within an order of magnitude" is the usual rule for any point estimates.
- The pool of data for the numerical methods is surprisingly small. Perhaps the best open database is CORE-DATA, held at the University of Birmingham, UK (Taylor-Adams and Kirwan, 1995; Gibson, 1998). Even this has only about 400 data points, from a limited number of industries.
- The techniques are predicated on the individual; there is no established technique for looking at teamwork errors or for the system-of-systems level.
- Almost none of the techniques take sufficient account of dependency; they assume each action within a procedure occurs in isolation from previous actions.
- While we can continue to produce answers required for the design phase, this is clearly an area

demanding further contributions from the human factors profession.

3.3.5. Workload assessment

There are a number of widely used subjective methods for assessing workload; for example the Borg scale (Borg, 1998), NASA TLX, and the SWAT (Vidulich and Tsang, 1986). These can be augmented by various physiological methods, as summarised, for example, in Hancock et al. (1985). However, there is a basic assumption in these that it is an individual's workload that is to be assessed, and that this assessment should measure instantaneous workload. What is missing and would need to be considered by the systems integrator, is assessment of workload distributed over teams, and workload assessment distributed over longer periods of time, such as the hour, the week and the year. Unfortunately, we have very weak tools for the latter.

3.3.6. Allocation of functions, and the creation of teams

This, too, is a difficult area for practitioners. While there is an extensive literature on the subject (e.g. Fitts, 1962; Jordan, 1963; Chapanis, 1965; Price, 1985; Smith and Carayon, 1995; Beevis and Essens, 1996), and a variety of methods for performing the allocation of functions (e.g. Wirstad, 1979; Mumford, 1983a; Price, 1985; Kantowitz and Sorkin, 1987; Checkland and Scholes, 1990; Konz, 1990; Dobson, Martin et al., 1991; Majchrzak and Gasser, 1992; Mital, Motorwala et al., 1994a; Mital, Motorwala et al., 1994b; Grote, Ryser et al., 2000) there are evident difficulties, arising from the earlier discussion about complexity and immortality. Firstly, not many of the methods proposed cover the allocation of functions between teams and technology. Industrial engineering techniques are available for this, as described in Konz (1990), but none of the methods are particularly good where cognitive tasks are concerned. The authors are not aware of any technique that adequately considers the 'human as hero' paradigm.

There are other issues as well. As Amalberti (2001) has argued, progressive introduction of safety regulations within an organisation can constrain the freedom of humans and teams to act, and by introducing changes to the technology can affect the allocation of functions. He presents a convincing argument that in fact this may be detrimental to safety, due to changes in personnel and career horizons. We know of no technique in human factors/ergonomics that adequately address this issue. There is a further cause not addressed by Amalberti; the effect of individuals who should be considered to be part of the team, but are not recognised as such. For example, in a given process the personnel from support processes such as health and safety, maintenance, and so one can, because of limited views of their responsibilities, make adjustments to a process that in fact make it more dangerous. This is an organisational structure problem ('who is responsible for what?'), allied to an organisational knowledge configuration problem ('who should know what?'), and is endemic in industry. About the only way in which this is addressed at the present time is through the design of high-reliability organisations (HROs), which place a heavy emphasis on culture to avoid problems and to improve the operational milieu (Weick, Sutcliffe et al., 1999; Reason, 2000, 2001; Weick and Sutcliffe, 2001).

Finally, we have neither adequate tools nor suitable metrics for predicting the future success (or not) of different organisational structures, whether they are teams or business units. The human factors/ergonomics literature provides no accepted means to answer questions such as "If I put together this team of people in these roles with these resources, what is the probability they will deliver a quality product within 1 year? And, given the team is created, how soon could I predict that failure is likely?" These questions are those that organisations face regularly, and, apart from ad hoc Monte-Carlo simulations, our answers are perhaps best expressed with a pause and a mumble about participatory organisational design.

3.3.7. Extended situation awareness

Situation awareness as described in the literature is a concept in need of extension. A very good, concise summary of orthodox thinking will be found in Jeannot (2000), with design guidance for the development of situation awareness in Endsley, Bolté et al. (2003). However, situation awareness in its current formulation is typically directed at the pilot's cockpit, with extension to the Air Traffic Control scenario. While this is important, the concept needs to be extended so that it is closely integrated with workload assessment (not just the instantaneous workload, but also the long-term, perhaps stressful workload). Furthermore, there is a need for more understanding of the team-working scenario; Salas, Prince et al. (1995), Sandom (2001) and Endsley, Bolté et al. (2003) for example discuss many of the issues (such as the need for communications so that all concerned understand the evolving situation even if they are not currently involved in it), the problems of different people having different interpretations, and so on, but designing for changes in team membership, or to cope with absenteeism, are not. Nor is historical awareness—the effect of small changes to the system by different people over long periods (Siemieniuch and Sinclair, 2004a) incorporated into the concept, and this can lead to disaster. Endsley, Bolté et al. (2003) is a good start to the design issues, but more is needed.

3.3.8. Knowledge lifecycle management

There are some excellent books on this, as the field stabilises; Stewart (1997), Boisot (1998), Davenport and

Prusak (1998) and Bontis, Dragonetti et al. (1999) are examples of these. From an ergonomics/human factors perspective, a number of other considerations will be found in Siemieniuch and Sinclair (2002, 2004b). However, the literature is still a little sparse on topics such as organisational knowledge configurations (who should know what?) and on tacit knowledge management and corporate memory. Currently, the situation seems to be that the configuration is determined by the knowledge and skills that can be identified from the processes on which an individual works (but see above on indeterminacy and incompleteness), and a reliance on 'experience' and peer groups to address the tacit knowledge dimension. However, this does not cope adequately with the problem; for example, it has been said that in the battlespace scenario that the training of air defence controllers is a process spanning some 25 years because of the need for a wealth of tacit knowledge to be gained. Much the same has been said for process managers in charge of complex production lines; if indeed this is the case, then we find ourselves having to plan the careers of personnel in order to fulfil the role requirements. Thus, knowledge lifecycle management becomes entwined with human resource management; it is not clear that the human factors/ergonomics profession has internalised this fully.

3.3.9. Governance issues

This is best summarised as 'are we doing the right things, and are we doing those things right?' This deals with organisational purposefulness, integrity, and probity in delivering good corporate and engineering governance. There are two aspects for system integration that should be considered; firstly, what is the best responsibility structure for good governance? The academic and business literature abounds with discussions of this at Board level, but there is next to nothing concerning such things as engineering governance below that level. While safety cases inevitably impinge on this, they do not specifically address this issue which is, essentially, long-cycle situation awareness (Siemieniuch and Sinclair, 2004a). Secondly, there seems to be nothing that answers the question, 'how much governance is enough?' More pertinently, this could be expressed as 'Where are the boundaries between disorder, good governance, and regimentation?' The pervasiveness of information technology throughout the organisation now means that, if they wished, the Board could micro-manage all aspects of the organisation's processes. Self-evidently, this is unwise, but all the notorious disasters (e.g. Chernobyl, Tenerife, Bhopal, Three Mile Island, Esso Longford, etc.) have poor governance within the organisation as a major contributing cause. So, we are left with a problem; what (and how many) are the 'golden rules' for governance, and how should they be exercised? These are issues with which systems integrators will have to grapple, without much advice available.

4. Conclusion

Across all domains there is an exponential growth in the complexity of the systems that comprise our industrialised society. We need to achieve a step-change in knowledge of how to design, integrate, operate and evolve systems that (a) are not fully understood by all stakeholders, (b) whose behaviour may not be fully predictable and (c) which function in an environment which cannot always be controlled. As stated in part 1 of the paper complexity is an inherent feature of these systems and is characterised by

- a (usually large) number of (usually strongly) interacting individual components of the system, and (probably) evolving interaction between the system and its environment, and
- the requirement for the system to adapt to change in a way that does not have adverse effects on the system's usefulness nor its ability to operate within a defined envelope of appropriate measurable parameters.

Although we often have in-depth expertise and knowledge about the characteristics and behaviour of individual system building-block components, there is a lack of understanding of the multiple, non-linear connections and dependencies among components—and of the way that they may self-organise, or co-evolve within a constantly changing environment over its lifecycle. The response of the complex system therefore cannot be adequately planned, understood, or anticipated. Understanding these issues is an incremental process and developing tools and approaches to aid the integration of hierarchies of components, sub-systems etc. and which will at least allow the properties of a complex system of systems to be bounded, would be a valuable first step.

The target end goal could be described as being able to deliver and manage systems with increasing complexity and with a high degree of uncertainty in the system operating space:

- with desired emergent properties
- in reduced timescales with controlled costs
- with improved flexibility, interoperability and supportability

through the development of the capability to engineer and integrate complex systems of systems:

- process, methods and techniques

- engineers with the required competence
- senior managers with the required vision

where complex systems may be characterised by one or more of the following attributes:

- Many and varied sub-system structures
- Many goals and objectives
- Many linkages/interactions between sub systems
- Low system transparency
- Many and varied internal and external interactions
- High ability to be self sustaining
- High likelihood of emergent properties
- High level of system intelligence
- High level of hierarchical organisation
- High degree of automony
- High level of robustness

To do this we need to identify the barriers that affect our ability to understand and model complex systems and their emergent behaviours and devise tools and methods to address issues such as

- a framework and language for addressing these issues (i.e. an ontology for complexity),
- the evolution and management of requirements and system configuration during design, maintenance, upgrades and operation of the system over its lifecycle
- the way that systems may self-organise, or co-evolve within a constantly changing environment, and still meet agreed, temporally sensitive goals,
- the way that systems may create new order—which may not be predictable—and communicate this to other systems (including the designers),
- factors which influence the emergence of undesirable emergent behaviours, and in the means and methods to predict and/or detect and filter out such emergent behaviour,
- IT aspects of visualisation and modelling/simulation of complex system,
- factors which constraint the hierarchical organisation of realisable systems

Although this paper has taken an ergonomics/human factors perspective on the issues raised, it is hoped that the reader will recognise that addressing any of the topics/problem areas outlined will require an integrated multi-disciplinary approach. The authors have attempted to provide one viewpoint onto what is, of itself, a hugely complex area and provide some insights into how some of the problems could be tackled. In terms of achieving true engineering excellence, effective systems integration is rapidly becoming the next holy grail.

References

Ahern, D.M., Clouse, A., Turner, R., 2003. Ergonomics of Human-System Interaction—Human-Centred Lifecycle Process Descriptions. Addison-Wesley, New York.

Amalberti, R., 2001. The paradoxes of almost totally safe transportation systems. Saf. Sci. 37 (2–3), 109–126.

Ashby, W.R., 1962. Introduction to cybernetics. In: Foerster, H.V., Zopf, G.W. (Eds.), Principles of Self-organising Systems. Pergamon, New York.

Atkins, P.W., 1986. The second law. W.H. Freeman, San Francisco.

Beevis, D., Essens, P., 1996. State-of-the-Art Report: Improving function Allocation for Integrated Systems Design. ALLFN'97—Revisiting the Allocation of Functions Issue, Galway. IEAPress, Louisville.

Boisot, M.H., 1998. Knowledge Assets: Securing Competitive Advantage in the Information Economy. Oxford University Press, Oxford.

Bontis, N., Dragonetti, N.C., et al., 1999. The knowledge toolbox: a review of the tools available to measure and manage intangible resources. Eur. Manag. J. 17 (4), 391–402.

Booher, H.R., 2003. Handbook of Human Systems Integration. Wiley Interscience, New York.

Borg, G., 1998. Borg's Perceived Exertion and Pain Scales. Human Kinetics, Champaign, IL.

Chapanis, A., 1965. On the allocation of functions between men and machines. Occup. Psychol. 39, 1–11.

Chapanis, A., 1996. Human Factors in Systems Engineering. Wiley-Interscience, New York.

Checkland, P.B., Scholes, J., 1990. Soft systems methodology in action. Wiley, Chichester, UK.

Chua, C.K., Leong, K.F., et al., 2003. RAPID PROTOTYPING: Principles and Applications. World Scientific Publishing Co. Pte. Ltd, Singapore.

Churchman, C.W., 1968. The Systems Approach. Delacorte Press, New York.

Cilliers, P., 1998. Complexity and Postmodernism. Routledge, London.

Crosby, P.B., 1979. Quality is Free: The Art of Making Quality Certain. McGraw Hill, New York.

Davenport, T.H., Prusak, L., 1998. Working Knowledge: How Organisations Manage What They Know. Harvard Business School Press, Boston.

deMeyer, A., Loch, C.H., et al., 2002. Managing project uncertainty: from variation to chaos. Sloan Manag. Rev. 43 (2), 60–67.

Dobson, J.E., Martin, M.J., et al., 1991. Determining Requirements for CSCW: The ORDIT Approach. Collaborative Work, Social Communications and Information Systems. Elsevier Science Publishers B.V., Amsteradm.

Dougherty, E.M., 1990. Human reliability analysis—where shouldst thou turn? Reliability Eng. Syst. Saf. 29, 283–299.

Eason, K.D., 1988. Information Technology and Organisational Change. Taylor & Francis, London.

Eason, K.D., Harker, S., 1991. Human factors contributions to the design process. In: Shackel, B., Richardson, S.J. (Eds.), Human Factors for Informatics Usability. CUP, Cambridge, pp. 73–96.

Eason, K.D., Harker, S.D.P., et al., 1997. Working with users to generate organisational requirements: the ORDIT methodology. ICL Syst. J. (January), 205–222.

Ehn, P., 1988. The Work-Oriented Design of Computer Artifacts. Arbetsmiljo, Arbelistraum, Stockholm.

Embrey, D.E., Humphreys, F., et al., 1984. SLIM-MAUD: An Approach to Assessing Human Error Probabilities Using Structured Expert Judgement. Nuclear Regulatory Commission.

Emery, F.E., 1969. Systems Thinking. Penguin, London.

Emery, F.E., Trist, E.L., 1960. Socio-Technical Systems. Pergamon Press, Oxford.

Endsley, M.R., Bolté, B., et al., 2003. Designing for Situation Awareness: An Approach to User-Centred Design. Yaylor & Francis, London.

Fitts, P.M., 1962. Functions of men in complex systems. Aerosp. Eng. 20 (1), 34–39.

Forsberg, K., Mooz, H., 1992. The relationship of systems engineering to the project cycle. Eng. Manag. J. 4 (3), 36–43.

Forsberg, K., Mooz, H., 1995. Application of the "vee" to incremental and evolutionary development Conference. Proceedings of the National Council for System Engineering, St. Louis.

Gibson, W. H., 1998. Development of the CORE-DATA Database in Collection and Classification of Human Reliability Data for Use in Probabilistic Safety Assessments, International Atomic Energy Association.

Grote, G., Ryser, C., et al., 2000. KOMPASS: a method for complementary function allocation in automated work systems. Int. J. Hum.-Comput. Stud. 52, 267–287.

Hancock, P.A., Meshkati, N., Robertson, M.M., 1985. Physiological reflections of mental workload. Aviat. Space Environ. Med. 56 (11), 1110–1114.

Herrmann, T., Hoffmann, M., et al., 2004. A modelling method for the development of groupware applications as socio-technical systems. Behav. Inform. Technol. 23 (2), 119–135.

Hippel, E.V., 1990. Task partitioning: an innovative process variable. Res. Policy 19, 407–418.

Hollnagel, E., 1998. Cognitive Reliability and Error Analysis Method. Elsevier, London.

Jeannot, A., 2000. Situation awareness, synthesis of literature research. Bretigny-sur-Orge, EUROCONTROL Experimental Centre.

Jordan, N., 1963. Allocation of functions between man and machines in automated systems. J. Appl. Psychol. 47 (3), 161–165.

Kantowitz, B., Sorkin, R., 1987. Allocation of functions. In: Salvendy, G. (Ed.), Handbook of Human Factors. Wiley, New York, pp. 355–369.

Kirwan, B., 1994. A Guide to Practical Human Reliability Assessment. Taylor & Francis, London.

Konz, S., 1990. Work Design: Industrial Ergonomics. Publishing Horizons, Scottsdale, Arizona.

Leonard, A.D., 1984. Technical notes on cybernetics. In: Clemson, B. (Ed.), Cybernetics: A New Management Tool. Abacus/Gordon & Breach, New York, pp. 199–224.

Lundgren-Cayrol, K., 2000. Cybernetics concepts, Concordia College. 2004.

Machol, R.E., 1965. Systems Engineering Handbook. McGraw-Hill, New York.

Majchrzak, A., Gasser, L., 1992. HITOP-A: a tool to facilitate interdisciplinary manufacturing systems design. Int. J. Hum. Factors Manuf. 2 (3), 255–276.

McConnell, S.C., 1996. Rapid Development: Taming Wild Software Schedules. Microsoft Press International.

Mital, A., Motorwala, A., et al., 1994a. Allocation of functions to humans and machines in a manufacturing environment: Part 1—Guidelines for practitioners. Int. J. Ind. Ergon. 14 (1 and 2), 3–31.

Mital, A., Motorwala, A., et al., 1994b. Allocation of functions to humans and machines in a manufacturing environment: Part 2—scientific basis (knowledge basis) for the guide. Int. J. Ind. Ergon. 14 (1 and 2), 33–49.

Moray, N., 1986. Monitoring behaviour and supervisory control. In: Boff, K.R., Kaufman, L., Thomas, J.P. (Eds.), Handbook of Perception and Performance, Vol. 2. Wiley, Ney York pp. 40.1–40.51.

Mumford, E., 1983. Participative systems design: practice and theory. J. Occup. Behav. 4, 47–57.

O'Hanlon, M., 2000. Technological Change and the Future of Warfare. Brookings Press, Washington.

Perrow, C., 1999. Normal Accidents—Living with High-Risk Technologies. Princeton University Press, Princeton, NJ.

Pocock, S., Harrison, M., et al., 2001. THEA: A Technique for Human Error Assessment Early in Design. Human-Computer Interaction: INTERACT-01. IOS Press.

Price, H.E., 1985. The allocation of functions in systems. Hum. Factors 27, 33–45.

Rasmussen, J., 1986. Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering. North-Holland, New York.

Rasmussen, J., Goodstein, L. P., 1986. Decision support in supervisory control. Analysis, design and evaluation of man-machine systems, Varese, Italy, Second IFAC/IFIP/IFORS/IEA Conference.

Reason, J., 2000. Human error: models and management. Br. Med. J. 320, 768–770.

Reason, J., 2001. The dimensions of organisational resilience to operational hazards. British Airways Human Factors Conference: Enhancing Operational Effectiveness, No publisher.

Rhodes, R., 2000. Visions Of Technology: A Century Of Vital Debate About Machines Systems And The Human World. Simon & Schuster, New York.

Salas, E., Prince, C., et al., 1995. Situation awareness in team performance: implications for measurement and training. Hum. Factors 37, 123–136.

Sandom, C., 2001. Situation awareness. In: Noyes, J., Brandsby, M. (Eds.), People in Control: Human Factors in Control Room Design, vol. 60. Institute of Electrical Engineers, London, pp. 51–68.

Sherwood-Jones, B., 2003. Human-centred design: opportunities and resources. HSIAC Gateway 14 (3), 1–4.

Sherwood-Jones, B., Earthy, J., et al., 2003. Human-Centred Design in a Systems Engineering Context. Annual Conference of the Ergonomics Society. Taylor & Francis, Edinburgh.

Siemieniuch, C.E., Sinclair, M.A., 2000. Implications Of the supply chain for role definitions in concurrent engineering. Int. J. Hum. Factors Ergon. Manuf. 10 (3), 251–272.

Siemieniuch, C.E., Sinclair, M.A., 2002. On complexity, process ownership and organisational learning in manufacturing organisations, from an ergonomics perspective. Appl. Ergon. 33 (5), 449–462.

Siemieniuch, C.E., Sinclair, M.A., 2004a. Long-Cycle, Distributed Situation Awareness and the Avoidance of Disasters. Human Performance, Situation Awareness and Automation: Current Research and Trends, Daytona Beach, USA, Lawrence Erlbaum Associates Mahwah, NJ.

Siemieniuch, C.E., Sinclair, M.A., 2004. Organisational readiness for knowledge management. Int. J. Oper. Prod. Manag. 24 (1), 79–98.

Singleton, W.T., 1974. Man-Machine Systems. Penguin, London.

Skyttner, L., 1996. General Systems Theory—An Introduction. McMillan, New York.

Smith, M.J., Carayon, P., 1995. New technology, automation, and work organisation: stress problems and improved technology implementations. Int. J. Hum. Factors Manuf. 5 (1), 99–116.

Stewart, T.A., 1997. Intellectual Capital: The New Wealth of Companies. Doubleday, New York.

Stolovitch, H.D., Keeps, E.J. (Eds.), 1999. Handbook of Human Performance Technology: Improving Individual and Organizational Performance Worldwide. Jossey-Bass/Pfeiffer, San Francisco.

Swain, A.D., Guttman, H.F., 1980. Handbook of Human Reliability Analysis with the Emphasis on Nuclear Power Plant Applications. Sandia Laboratories, N.M.

Taylor-Adams, S.E., Kirwan, B., 1995. Human reliability data requirements. Int. J. Qual. Reliability Manag. 12 (1), 24–46.

Vicente, K., 1990. Coherence- and correspondence-driven work domains; implications for systems design. Behav. Inform. Technol. 9 (6), 493–502.

Vicente, K.J., 1998. Human factors and global problems: a systems approach. Syst. Eng. 1 (1), 57–69.

Vicente, K.J., Burns, C.M., et al., 1997. Muddling through wicked design problems. Ergon. Des. 5 (1), 25–30.

Vicente, K. J., Rasmussen, J., 1988. On Applying the Skills, Rules and Knowledge Framework to Interface Design. Engineering Psychology Research Laboratory, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana IL 61801.

Vidulich, M.A., Tsang, P.S., 1986. Techniques of subjective workload assessment: a comparison of SWAT and NASA-Bipolar methods. Ergonomics 29, 1385–1398.

Weick, K.E., Sutcliffe, K.M., 2001. Managing the Unexpected : Assuring High Performance in an Age of Complexity. Jossey-Bass, San Francisco.

Weick, K.E., Sutcliffe, K.M., et al., 1999. Organizing for high reliability: processes of collective mindfulness. Res. Org. Behav. 21, 23–81.

Williams, J.C., 1985. Validation of human reliability assessment techniques. Reliability Eng. 11, 149–162.

Wirstad, J., 1979. On the Allocation of Functions Between Human and Machine—Analysis of Concept and its Implication for Control System Ergonomics. Ergonområd, Karlstad, Sweden.