Chapter 5 – ARTIFICIAL COGNITION AND CO-OPERATIVE AUTOMATION

Chapter Lead: A. Schulte

Contributors: M. Chamberlin, J. Edwards, A. Schulte, R. Taylor, M. Waters

Within the scope of this report on "Uninhabited Military Vehicles: Human Factors in Augmenting the Force", the present chapter is dedicated to the involvement of the human factor with the specific aspect of the integration of artificial cognition in the process of vehicle guidance and supervision. In particular, the idea of co-operative control, i.e., the co-operation between the human operator and automation, will be addressed. Hence, human-automation integration can be viewed from two different standpoints, each of which facilitating the other. On the one hand, the human has to be considered as the user of technology, being the operator in a somehow automated work environment, responsible for the pursuit of the ongoing processes and provided with more or less authority. On the other hand, the consideration of human performance in work processes suggests unique approaches to automation and decision systems design for the future. These approaches reveal the potential of human-like behaving machines (in the sense of rational behaviour) in certain given task domains, even being able to co-operate, as well as the potential of a human-centred automation, promising significant performance advances, once introduced into a work place.

The following sections will provide a discussion of the human involvement aspects as named above from a conceptual point of view, to begin with. Further down, application examples taken from current research will be illustrated, covering different application areas as well as different perspectives in terms of human involvement.

Firstly, the scope of the discussion will be delimited. Bearing in mind that the following considerations shall have the potential to be applicable in the air, land, see, space and underwater domain likewise, it is useful to restrict oneself to some more specific field, in particular in closely application-related research. So, the aviation domain, specifically flight guidance and mission management of military aircraft, conventionally manned and unmanned likewise, will mark the vantage point of the following discussion. The motivation of this selection against the background of the consideration of human cognition and decision-making will be explained.

The next section will provide a statement on the current, i.e., the solution of conventional automation being strongly influenced by the paradigm of supervisory control. A framework for the modelling of the work process and related control levels will be briefly discussed. Domain specific technology approaches will be roughly structured, again considering flight guidance as an example.

Problems arising from conventional automation approaches will be discussed in the third section. Perspectives of future automation and required extensions will be introduced. At this stage the notion of an Artificial Cognitive Unit (ACU) as part of a work system will be introduced. The required capabilities of such a machine being a team mate will be estimated.

The outcome of the consideration of these required advances in automation is to concentrate on the treatment of human and machine cognition as an inter-disciplinary approach based upon cognitive psychology and artificial intelligence as branch of information technology. This will be the objective of the fourth section. As an interim result the theory of the Cognitive Process will be introduced in this section.

Section 5.5 briefly gives some information on realisation aspects of the Cognitive Process, being the underlying theory itself. Creating a cognitive system according to this theory requires the implementation of a systems engineering framework.

Section 5.6 will broaden the view from which the issue of artificial cognition and co-operative automation has been looked at so far, by opening up the podium for different, but related perspectives covering the fields of Artificial Intelligence methods evaluation, knowledge engineering and an application in the underwater vehicle guidance domain.

5.1 SCOPE

As already mentioned in the introduction the rather broad scope of possible air, land, sea, space and underwater applications needs to be narrowed somehow, taking advantage of digging deeper into the specific problems of one particular domain, finally providing beneficial insight ready to be adopted by other application areas. In anticipation of the main scope of this chapter the airborne application is the choice. It is a fact, that conventional automation, a term which will be defined further down, can be regarded as very advanced in this domain. Modern electronic fly-by-wire systems enable an almost fully automatic performance of an entire mission, as daily demonstrated in thousands of civil airliner flights. Generating and pursuing a four-dimensional flight trajectory is not a real technological challenge any more, but provides a very sustainable platform for further considerations to be endeavoured here. Especially the higher levels of cognitive performance involving problem-solving, and decision-making are still mostly attributed to the human operator acting as supervisor of a technical process.

In contrast to this, the situation, e.g., in ground based application is somehow inverted. Autonomous driving is still a complicated issue (as observed during the recent DARPA grand challenge), i.e., the automation of the lower guidance levels including the recognition of the nearest environment and the resulting stabilisation and tracking tasks are not at all fully available today. In fact, current research is focused here. On the other hand a car navigation system supporting on the supervisory control level is almost present in every upper middle-sized class car. Virtually every tactical decision emerging in every day's driving will already be covered. Currently up-coming so-called driver assistant systems, which in most cases correspond with the functions of conventional aircraft automation, call for action in terms of central co-ordination of their supervision for efficient operation.

Again, the scope of this chapter shall be the automation of tasks on that supervisory level. As a result, systems shall be enabled towards autonomous task accomplishment. This issue of autonomy will be discussed in some more depth. Another very important issue will be the consideration of human-machine teaming and co-operation. The following sub-sections outline a typical aerial warfare mission to serve as a benchmark, providing a most interesting challenge for the concepts to be presented here – i.e., the relevant scenario shall be described mostly on a symbolic level, minimising the involvement of the processing of signals. The focus shall be more upon the logical relations between the objects, rather than upon their physical properties.

5.1.1 Typical Scenario from the Military Aviation Domain

The benchmark mission shall be taken from the aerial warfare domain. Figure 5-1 depicts an overview of the scenario and the relevant objects of a multi-ship air-to-ground attack mission. The own forces consist of the airborne component covering different rolls such as reconnaissance (RECCE), suppression of enemy air defence (SEAD) and attack. Furthermore, a command and control (C2) component might be involved, possibly airborne, typically ground-based.

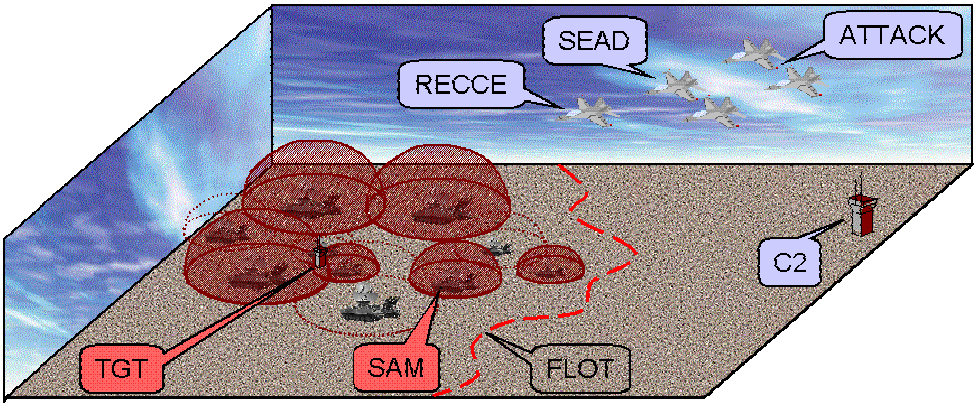


Figure 5-1: Scenario for Multi-Ship Air-to-Ground Attack Mission.

The hostile forces consist mainly of two components, i.e., a military target, fixed or moving and a ground-based air-defence system represented by surface-to-air missile (SAM) sites, which can be switched on and off, and which are to some extent known during the mission preparation phase both of which are separated from the safe territory by the forward line of own troops (FLOT). The mission order requires the attack component to destroy the hostile target. To achieve this, SAM-sites temporarily have to be suppressed or destroyed.

Although massively simplified with respect to asymmetric warfare scenarios currently discussed by NATO, this scenario bears a great variety of challenges in terms of integrated mission systems and automation.

5.1.2 Forces Structure

The own airborne forces will be a whatsoever mix of manned and un-manned platforms, to begin with. The scenario envisions a set of platforms, which have no static role or task allocation (A static role allocation in this context could be "Reconnaissance A/C or UAV searching, combat A/C or UAV shooting"). These platforms form a heterogeneous team, which means that the entities may differ from each other with respect to resources and capabilities, such as sensors, actuators, weapons, and information processing. This heterogeneous team structure does not prohibit homogeneous sub-structures, i.e., that some team members have equal or partially overlapping resources and capabilities. The envisioned scenario requires co-operation capabilities of the participating forces, because otherwise the mission cannot be accomplished [1].

A more generalised standpoint is shown in Figure 5-2. Starting from a classical situation, where single or multiple manned vehicles perform the mission. The critical questions arise when un-inhabited aerial vehicles (UAV) enter the scene. It has to be decided whether the UAVs will substitute or supplement the conventional manned platforms [2]. Obviously in some cases substitution will be an isolated solution, especially thinking of the so-called DDD-missions (dull-dirty-dangerous) – but generally, we certainly have to face the technological challenges of the solution of supplementation of forces, including the issues of manned-unmanned teaming, co-operation and supervision.

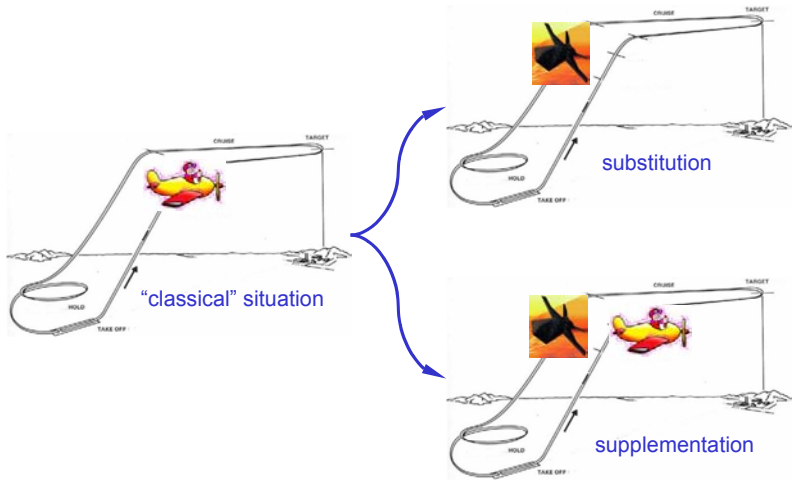


Figure 5-2: Possible Characteristics in Future UAV Deployment – Substitution and/or Supplementation.

5.1.3 References

[1] Ertl, C. and Schulte, A. (2004, September). System Design Concepts for Co-operative and Autonomous Mission Accomplishment of UAVs. In: Deutscher Luft- und Raumfahrtkongress. Dresden, GE. 20-23.

[2] Schulte, A. (2003, 10th – 13th June). Systems Engineering Framework Defining Required Functions of Un-inhabited Intelligent Vehicle Guidance. In: NATO RTO. Human Factors and Medicine Panel. Task Group HFM-078 on Unmanned Military Vehicles: Human Factors in Augmenting the Force. Leiden, NL.

5.2 THE WORK PROCESS AND CONVENTIONAL AUTOMATION'S SOLUTION

The last section gave a brief outline of the challenge for future mission systems. Needless to say, this type of mission can already be performed today, in one or the other way. The scope of this report of course is the augmented exploitation of presently unrevealed abilities in manned-unmanned teaming. To do so, the first step here shall be characterisation of current automation, i.e., the solution of conventional automation. For the later discrimination between automatic and autonomous performance the consideration of the work system will be helpful.

5.2.1 The Work System

The work system as a general ergonomics concept [1] has been utilised in the application domain of human-machine co-operation in aircraft flight guidance by [2]. Figure 5-3 shows an adaptation of the concept incorporating some application specific imagery for the purpose of intuitive understanding.

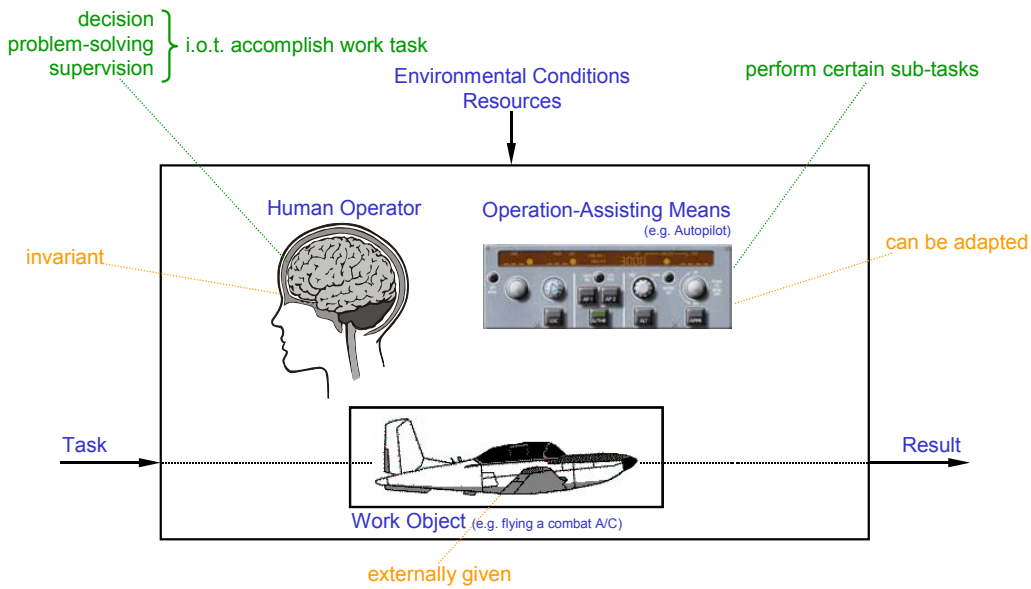


Figure 5-3: Concept of Work System.

The work system consists of three major elements, i.e., the operator, the work object and operation-assisting means, as characterised in some more detail here:

- **Operator:** In the traditional view of a work system the operator is usually a human operator, being in charge of performing a certain given task, such as accomplishing a combat mission, as according to the chosen application. The human operator is the high end decision element of the work system. He determines and supervises within the work system what will happen with the work object. This can be done by working on any required performance level, including manual control. In highly automated work systems, as we are talking of, the human performance is usually focused on supervisory control, including decision-making and problem-solving in order to comply with the work task. As according to the common view of ergonomics, the abilities of the skilled and trained human operator in terms of information processing performance can be seen as pretty much invariant in an average.
- **Work Object:** The notion of the work object is not necessarily restricted to the physical nature of whatever machine, but also comprises dynamical processes, i.e., the progression of the situation over time. In the chosen application domain, the work object may be the mission of a combat aircraft or UAV.
- **Operation-Assisting Means:** The concept of the operation-assisting means can be seen as a container for whatever tools or automation of the work place is available, being computerised pieces of technology in many cases. In our application domain an auto-flight/autopilot system including the human-machine control interface (i.e., FCU – flight control unit), or even the aircraft itself as a means of transport may serve as typical examples. Common to the nature of various operation-assisting means is the fact that they only perform certain sub-tasks (e.g., pursuing a given flight trajectory, holding a defined heading). Such a sub-task does not form a work system itself, obviously being only a part of another higher level work task. In today's common ergonomic design, the operation-assisting means are typically subjected to the endeavours of adaptation and optimisation in order to meet overall system requirements and further improvements.

These elements will be combined to the work system set up in order to achieve a certain work result on the basis of a given high level work task. The accomplishment of a military flight mission may give a good idea of what is meant here. Finally, environmental conditions and external resources, such as information, material, or energy will affect the ongoing work process.

The concept of the work system seems very suitable for the consideration of problems to be discussed in the further pursuit of this elaboration. The reason for this is the fact that the work process is constituted by the work task and the desired result, no matter its technical or organisational structure. However, exactly this technical or organisational structure might as well be easily modelled and analysed by the framework given by the work system. Yet, the notion of the work system at this stage gives no hints of modelling the mechanisms of human performance.

In order to do so, a very common model of human control performance shall be mentioned here, where a distinction is drawn between manual and supervisory control. This issue has been elaborately investigated by Thomas B. Sheridan at MIT (e.g., [3]) with a more recent focus on tele-operation [4], where obviously supervisory control predominates due to the remoteness of the work object.

Figure 5-4, which is adapted from [3], shows an automated human-machine system with the human operator in manual control mode on the left hand side. In this situation the human operator is busy in feedback control of the inner loops of the underlying process. Typical for the manual control mode are any kind of tracking tasks such as lateral car steering or attitude control of an aircraft. Automation is mainly responsible for the transformation and transmission of the required signals. On the right hand side of Figure 5-4, the automation takes over the role of automatically closing higher bandwidth control loops as to the dynamic process. In this case the human operator's role is shifted towards the supervisory control mode, where the tasks of monitoring and setting demand values for the automated control process are relevant.

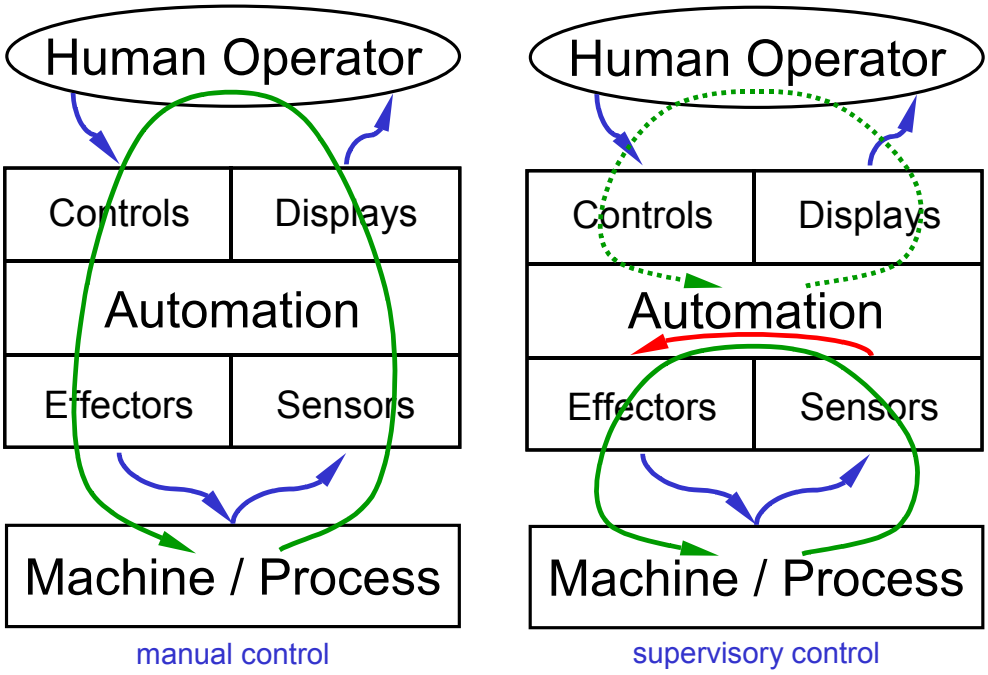


Figure 5-4: Manual and Supervisory Control.

In order to approach another definition of supervisory control [5] states:

> *"When a process is semi-automated or responds very slowly, it is not necessary for a human to devote full attention to that process, […] In situations where […the process] is automatically controlled, we can view the human as a supervisor whose role includes monitoring the process […], adjusting the reference points […], and intervening in the case of failures and emergencies." [Rouse, 1980]*

Many real-world applications in fact will require human-machine interaction as a mixture of manual and supervisory control as a function of the level of automation selected. The human operator will permanently toggle between the two control modes, allocating varying amounts of attention to one or the other task.

In order to prepare a common ground for the further discussion of models of human performance, this sub-section shall be closing with the introduction of a human model of manual and supervisory control advocated by [5]. The adapted model is set in the aviation domain context and depicted in Figure 5-5.

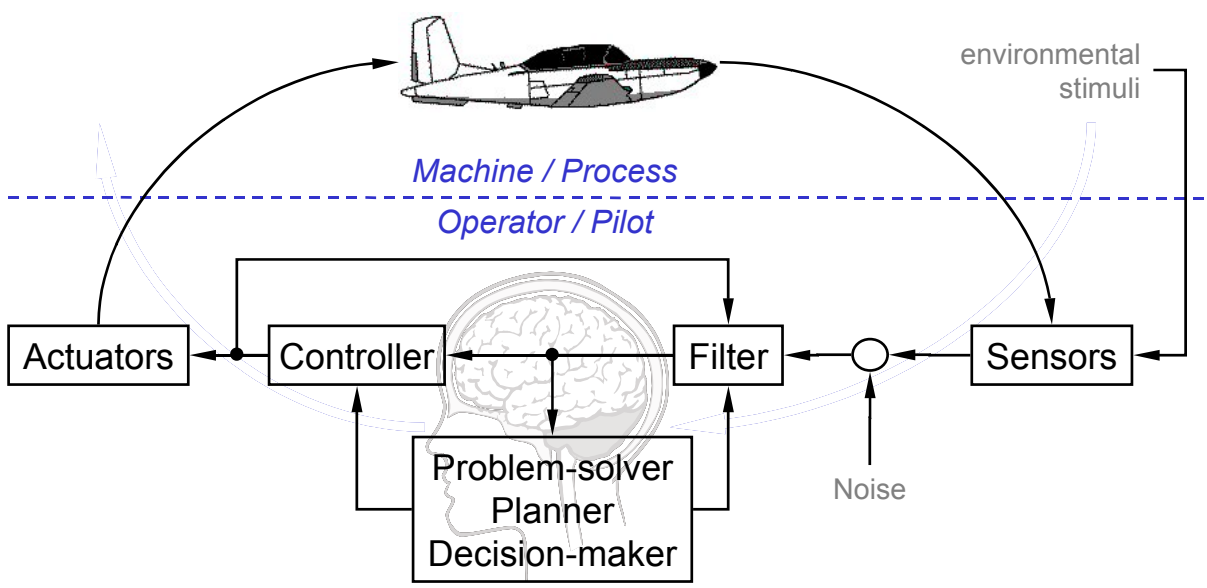


Figure 5-5: Model of Human Manual and Supervisory Control.

In Figure 5-5 the direct functional chain of sensing process and environmental parameters, filtering the information, applying control laws, and finally acting on the process represents all that is involved in the execution of manual control. On a supervisory control level gathered and filtered information will be fed into a functional block representing problem-solving, planning and decision-making. This block in turn will determine the demand values for the controller. Furthermore it allows the selection of the control mode and the adaptation of the control laws according to the current task. Finally, the decision-maker will adjust the filter in terms of selective allocation of resources such as attention (e.g., [6]).

5.2.2 The Hierarchy of a Conventional Guidance and Control System

In the previous sub-section the focus was drawn to the human operator's aspects of the work system for the first time. This sub-section shall concentrate more upon the operation-assisting means, i.e., the automation.

Obviously, the characteristic of the operation-assisting means is dependent on the application domain to a great extent. This is the point where we get back to the aviation domain as an example.

Figure 5-6 shows the major building blocks of a common hierarchical architecture of a state-of-the-art flight guidance and control system with several nested loops (adapted from [7]). Besides the many closed control loops on the machine side, one loop is closed involving the human operator, i.e., the pilot. Obviously this architecture puts the pilot into a versatile situation of supervisory control, using all these fancy machine functions. Direct intervention in manual control style is likewise possible on the lowest (i.e., most right in Figure 5-6) interaction level. It should be mentioned that Sheridan's notion of manual control, being unaffected by automated control loops, is to some extent impaired by the current technology of control-configured vehicles ("fly-by-wire"), where the lowest available human interaction level already implicates automatic control. Nevertheless, the human interaction with such a system might be denoted as manual control on that particular interaction level.

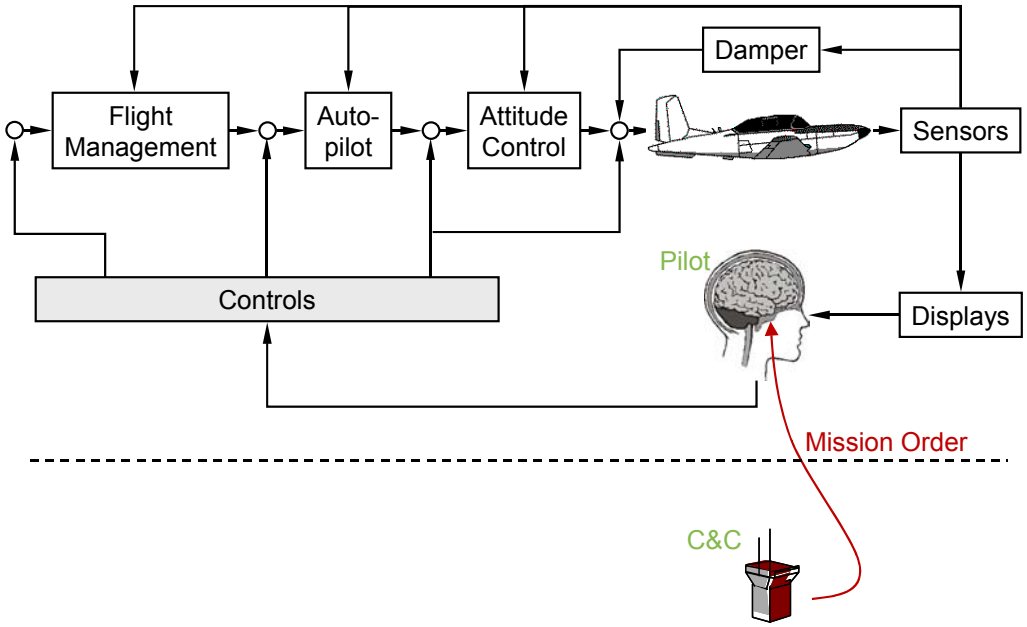


Figure 5-6: Conventional Guidance and Control System (Manned A/C).

While the pilot is controlling and supervising his machine, he himself is supervised by some external authority such as any imaginable implementation of command and control. In many western leaderships, the interface between command and control and the local operators is implemented on the basis of the assignment of work orders, i.e., mission orders in our domain.

A major performance feature of an educated, trained, and well skilled operator is the capability of transforming this work order into a desired work result. This structure, though, is tightly related to the conception of the work system according to the previous sub-section.

Figure 5-7 shows a situation which emerges when the pilot is removed from the vehicle and placed somewhere else, e.g., in a ground control station. Again, this remote operator will receive a mission order from any superior command and control authority. Usually, the operator now will interact with the UAV by

passing a detailed mission plan, which has to be worked out on the basis of the mission order, to the vehicle. In the case of a fully automated system, this initial mission plan will be pursued by the vehicle by use of the available on-board technology. Usually, with conventional technology, exceptional situations on the mission level, such as occurring obstacles, changes in the tactical situation, or other constraining factors cannot be handled. As a result of a monitoring function of the ground operator adaptations of the mission plan or reversing to outer loop guidance commands may occur. Usually, there are a couple of restraining factors for the remote operation of the vehicle:

- Manual control of the inner loops may not be possible or desirable because of intolerable time delays in the data transmission with respect to the inner loop dynamics time constants. Thus, the remote operation heavily relies upon the availability, the performance and integrity of some specific guidance functions, such as auto-land, otherwise requiring manual interactions.
- Insufficient downlink bandwidth and/or incomplete sensor coverage, with respect to the task, can cause what may be called "keyhole perspective" [8] for the remote operator, potentially affecting the correctness or quality of his or her decisions.
- The availability of data link, i.e., the ability to monitor (via telemetry) or control (via telecommand) the vehicle remotely may be disturbed. As a result, no recognition of nor reaction to unexpected situations is possible any more on the human operator's side.

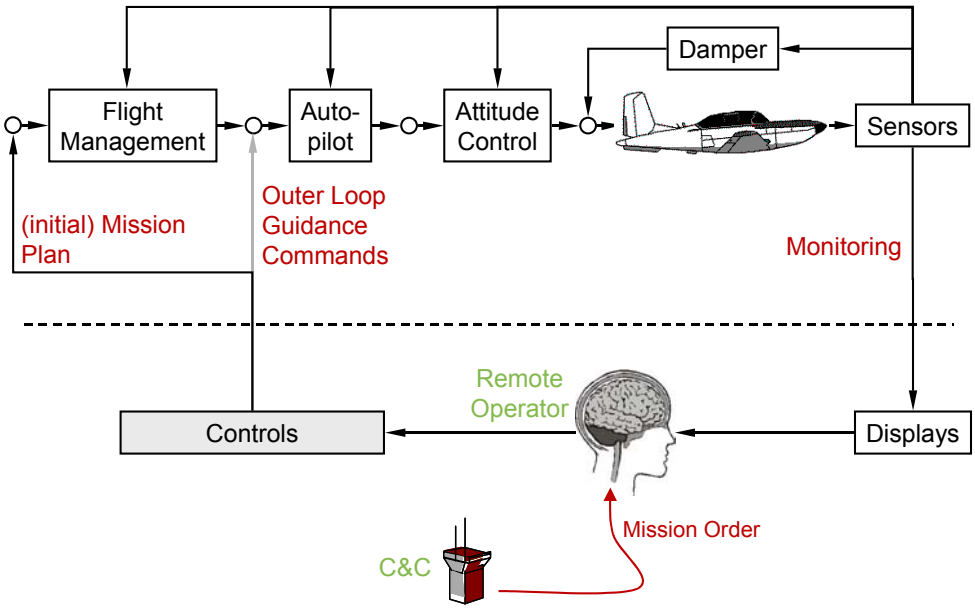


Figure 5-7: Conventional Guidance and Control System (Unmanned A/C).

What just has been elaborated for the flight guidance and navigation task holds true for other concurrent tasks of the operator, such as responding to a tactical environment or deploying mission related payload, as well. Air-to-air combat may serve as an extreme example, where sensory information from radar and identification equipment dictate the operator's actions with regard to trajectory determination as well as weapon aiming and deployment, altogether facilitated by complex, highly automated systems themselves. Here again a complicated mixture of manual and supervisory control tasks can be observed. Automation technology is predominantly available on the manual control level, if at all.

Figure 5-8 tries to summarise the just now characterised situation with respect to conventional automation. In order to achieve a desired work result, running a machine or controlling a process, usually a more or less wide spectrum of tasks and related sub-tasks has to be worked on. This may include sub-tasks such as flying an aircraft, operating in a tactical scenario, managing avionics systems, and communicating with others, each of which involving automation to some specific extent. Although there may be "horizontal" interaction between automation involved in different task domains to some limited extent (e.g., the automatic performance of a terrain evasive manoeuvre, or the automatic transmission of radar tracks via tactical data link), the integration of information in order to pursue the overall task is performed by the human operator on a supervisory control level mostly. So, the interaction within the automation is predominantly vertically structured. A conventional flight guidance and control system (see Figure 5-6) is certainly a very good example, whereas the human operator is supposed to toggle between the different tasks horizontally on a supervisory performance level.

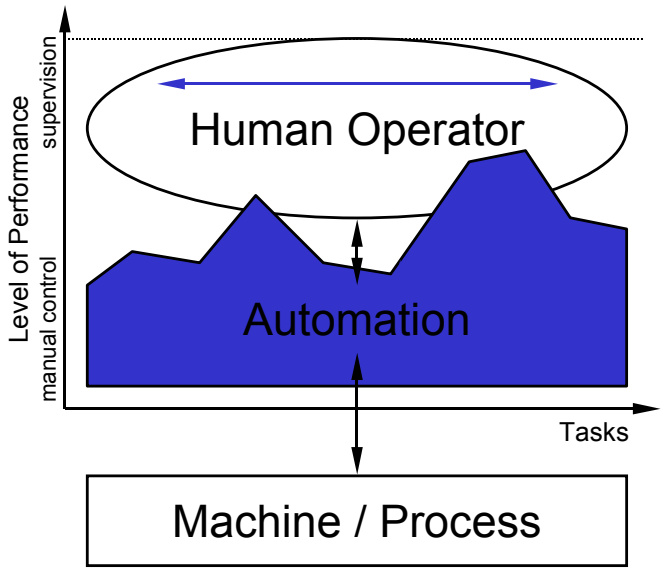


Figure 5-8: Organisational Structure of Conventionally, i.e., Hierarchically Automated Human-Machine Systems.

Having this rather simple organisational model of automation at hand the following section shall illuminate some technical pitfalls associated with this structure before some suggestions of improvements will be made.

5.2.3 References

[1] REFA (1984). (Verband für Arbeitsstudien und Betriebsorganisation e.V.). Methodenlehre des Arbeitsstudiums. Teil 1: Grundlagen. Hanser-Verlag.

[2] Onken, R. (2002, 7-9 October). Cognitive Cooperation for the Sake of the Human-Machine Team Effectiveness. In: RTO-HFM Symposium on The Role of Humans in Intelligent and Automated Systems. Warsaw, Poland.

[3] Sheridan, T.B. (1987). Supervisory Control. In: G. Salvendy (Ed.). Handbook of Human Factors. Chapter 9.6. pp. 1245-1268. John Wiley & Sons. New York.

[4] Sheridan, T.B. (1992). Telerobotics, Automation and Human Supervisory Control. MIT Press.

[5] Rouse, W.B. (1980). Systems Engineering Models of Human-Machine Interaction. Elsevier North Holland.

[6] Wickens, C.D. (1992). Engineering Psychology and Human Performance. Second Edition. HarperCollins Publishers.

[7] Brockhaus, R. (2001). Flugregelung. Zweite Auflage. Springer.

[8] Woods, D.D. (1984). Visual Momentum: A Concept to improve the cognitive coupling of person and computer. In: International Journal of Man-Machine Studies. 21, 229-244.

5.3 PROBLEM DEFINITION

The last section introduced one possible approach to how automation in human-machine systems could be looked at. Without being too specific on particular mission systems, some peculiarities of current, i.e., conventional automation systems have been deduced. This section provides a closer look upon problems which may arise in use of this automation approach. In the further pursuit of this section a possible perspective of future automation technology will be given, finally ending up with some very particular requirements to be implemented before such systems will be put into work.

5.3.1 Shortfalls with Conventional Automation

It has long since been known that erroneous human action is the predominating factor in aviation accidents, however, it is fair to state that many of these human errors are caused by over-demands (see grey line in Figure 5-9) on the pilot's resources [1], the latter representing the natural limiting factor for performance (see straight blue line in Figure 5-9). In order to overcome this situation the introduction of automation as described above was most beneficial in many situations, which otherwise could not be handled (see green line in Figure 5-9). On the other hand, new types of latent overtaxing-prone situations appeared with the increased introduction of automated functions [Onken, 1999] (see red line in Figure 5-9).

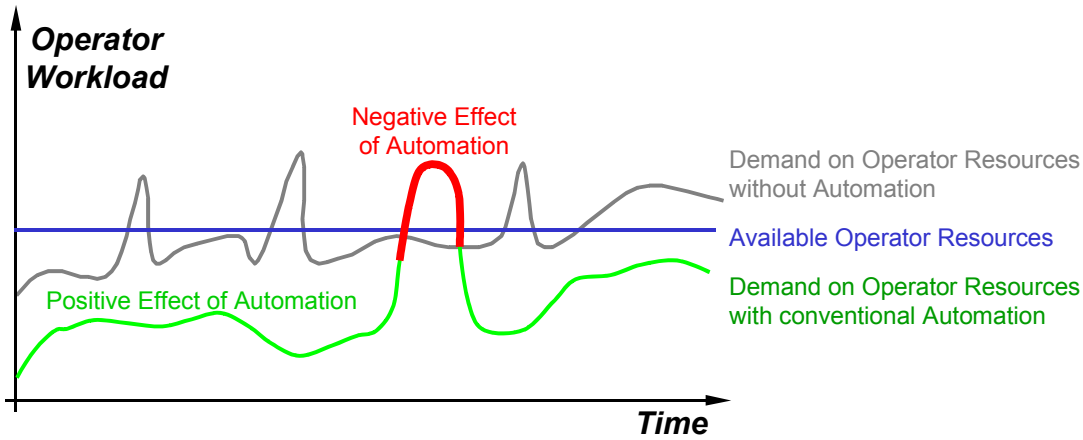


Figure 5-9: Operator Overload Caused by Conventional Automation.

Charles E. Billings investigated typical shortfalls of current aviation automation [2], with a particular view upon the human interaction with automation. According to Billings, the most critical design factors are complexity (Will the extent of the automatic function be fully understood by the human operator?), brittleness (Will the complex automation be fit for any imaginable situation or purpose?), opacity (Will the automatic execution provide sufficient and intelligible feedback to the human operator?), and literalism (Will the automation understand the human operator's control actions as 'naturally' as they are meant?). Generally spoken, Billings' answer to these questions with respect to current automation is "No", resulting in a situation which is usually referred to as clumsy automation [3].

Figure 5-10 explains the situation by use of the organisational structure of conventional automation with respect to task allocation between automation and the human operator as introduced previously (see Figure 5-8). Obviously, the classical task allocation suffers from some typical difficulties [23].

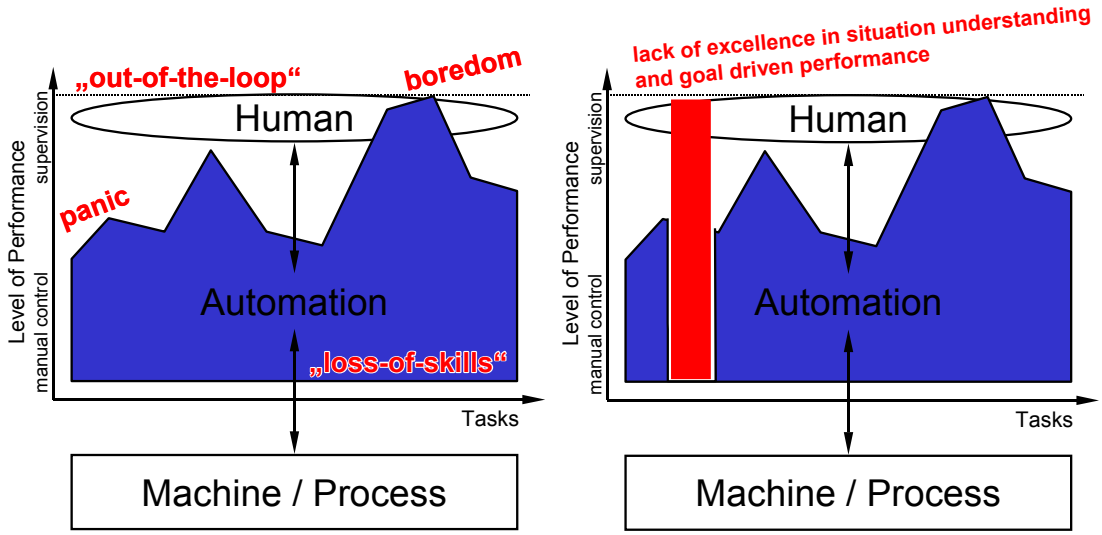


Figure 5-10: Shortfalls with Conventional Automation.

In particular under the assumption of increasing complexity of automation, the human operator is almost completely separated from the underlying process. The long term problem of loss of skills, i.e., erosion of competence, in supervisory control has been widely reported on, e.g., [4,5]. Within the same class of difficulties the human-out-of-the-loop problem represents the corresponding short term issue, addressing situations where operators almost fully rely upon the automation performance to an extent that any abnormal situation will inevitably cause human overload and erroneous action. [6] states:

> *"[...] by taking away the easy parts of his task, automation can make the difficult parts of a human operator's task more difficult." [6]*

Quite closely linked with Billings' notion of brittleness is the perception that conventional automation will usually not be able to recover from undesired situations induced by malfunctions, faulty operations or just the unexpected. The major reason for this limpness of the system is its lack of excellence in situation understanding and goal driven performance on the machine side, i.e., the missing capability of current automation systems to perform on a supervisory level in order to pursue the overall goals of the work system. As a good explanation for this circumstance the example of a simple autopilot function may serve [24].

Once activated, an "altitude acquire" function will pursue its specific sub-task of capturing a flight altitude pre-selected by the pilot in an almost perfect manner, no matter what may be of any relevance otherwise, e.g., ground or traffic proximity, exposure to enemy radar, or faulty demand setting or mode selection by the pilot in the sense of for instance a misinterpreted ATC clearance. So, automation offers a dedicated set of more or less independent functions, each of which being responsible for a particular sub-task. The situation can get even more precarious when these functions start getting linked horizontally without that being transparent to the human operator (i.e., opacity due to [Billings, 2]). Modern flight management systems often bear this characteristic, but still, conventional automation is not at all capable of performing any higher decision loop in the sense of supervisory control.

5.3.2 Perspectives of Future Automation

As an essence from the last sub-section, automation complexity can be seen as the most critical issue. To begin with, complex automation used to be the key to a major increase in mission effectiveness and flight safety (see Figure 5-11). Due to limited resources and capabilities on the human operator's side, a further increase of automation complexity has no longer been beneficial in terms of these productivity factors (see Figure 5-11). Obviously, automation became too complex to be reliably handled by human operators. The reason for this seems to be found in the unpredictability of the machine's behaviour due to inconsistencies between the machine function and the human operator's mental model of it. Conventional automation itself, in the first place meant to be an operation-assisting means, became a complex element within the already complex work system.

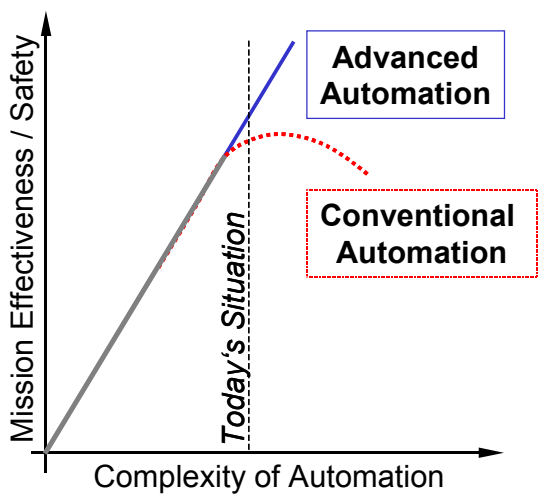


Figure 5-11: Perspectives of Future Automation.

In order to tackle this problem a new approach to automation has to be introduced into work systems. Figure 5-11 illustrates the vision of further increasing the productivity factors effectiveness and safety by advanced automation at the cost of furthermore complexity, but how shall this "advanced automation" be shaped?

Figure 5-12 tries to illustrate some first ideas in order to overcome the problems with conventional automation described earlier. Advanced automation shall not displace the human operator in a work system, but share the tasks in a close-partner work relationship. Task allocation shall not be static, but may be adapted to the current

situation's needs. This includes the facilitation of redundancy in functions in principal by at least a partial overlap in capabilities with respect to the task spectrum. The responsibility of automation (not necessarily authority) shall be extended to the supervisory control level, i.e., automation shall be enabled to perform certain tasks under consideration of the overall work task of the work system. Thereby, particularly brittleness will be tackled. Coordination and communication with such an automation system shall be supported on all performance levels, i.e., reaching from detailed low level information (reducing opacity of the machine solutions) up to abstract human-like information exchange on the supervisory level (tackling literalism of the automation). In general, it may be accepted that this approach to cognitive coupling [7] can be a contributing factor to the mitigation of disadvantageous complexity effects.

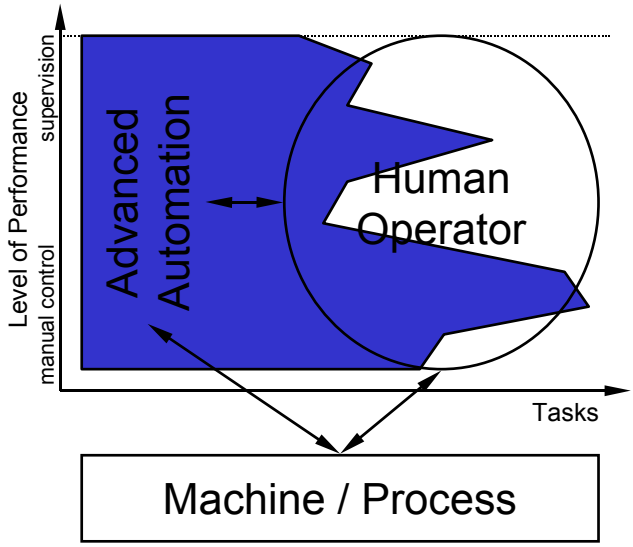


Figure 5-12: Co-operative Structure of Human-Machine Systems with Advanced Automation.

5.3.2.1 Cognitive Automation

An entity enabled to exhibit the aforementioned behaviour facets shall be referred to as Artificial Cognitive Unit (ACU) [24]. As indicated above, supervision and co-operation, as accomplishments of a machine system, require special capabilities. These capabilities were combined within the notion of such an Artificial Cognitive Unit. Obviously, the performance feature of cognition is the core element which has to be dealt with in order to design such an ACU. From the point of view of the discipline of cognitive psychology (e.g., [8,9]) human, i.e., natural cognition can be described by considering:

- Perception and allocation of attention;
- Knowledge representation and memory;
- Problem solving, reasoning and decision making;
- Language comprehension and its generation; and
- Learning and the development of expertise.

The availability of at least some of these aspects of cognition are the necessary pre-requisite to perform the supervisory control task (compare Sheridan, [26]) with respect to the compliancy with the overall work task.

Figure 5-13 shows the work system, according to Figure 5-3, with the human operator mimicked by an ACU. In this configuration the ACU represents all the performance requirements found to be attributed to the human operator earlier on, i.e., the performance of decision-making, problem-solving and supervision of the operation-assisting means and the work object in order to comply with the overall work task. The major difference is that the ACU is no longer invariant in terms of performance characteristics like its human archetype, but on the other hand, there has to be found a way how to design it according to the abovementioned requirements.

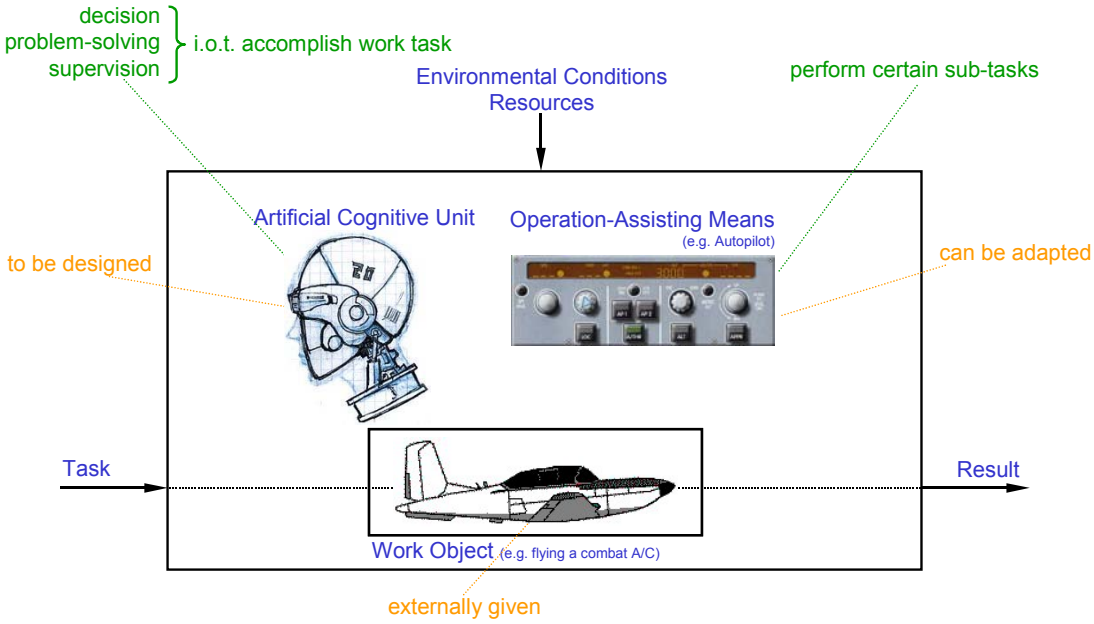


Figure 5-13: Artificial Cognitive Unit (ACU) Mimicking Human Operator in a "Work System".

Strictly, Figure 5-13 is not representing a work system any longer, since the presence of a human operator as part of the operating element is required by definition [25]. Such a system would be degenerated from the standpoint of "work", just existing for its own sake and not serving any human purpose. As soon as the human is involved as the tasking and monitoring element, which is always the case, the human will be part of the work system. The implications of this statement shall be discussed in the subsequent paragraph.

5.3.2.2 Automatic and Autonomous Performance

The (theoretical) configuration depicted in Figure 5-13, where the system is functioning (i.e., transforming the work object, e.g., flight, according to a work task into a desired work result) independently from any human intervention, can be referred to as being an autonomous system with respect to that particular work task. For this definition of autonomy a crucial factor is that a full work system is considered. Automated part-tasks, such as autopilot functions, working independently from human intervention likewise, are considered to be automatic.

Figure 5-14 [10] has to be understood in connection with the Figures 5-6 and 5-7. It shows the separation of automatic and autonomous systems from a more general point of view. The framed elements in Figure 5-14 form the considered work system.

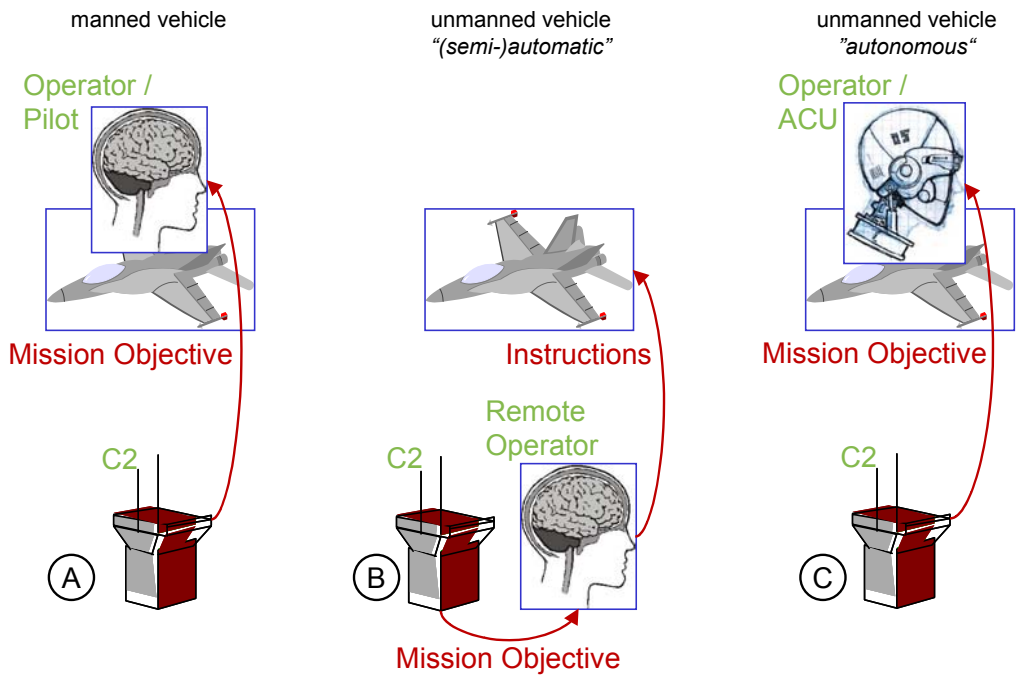


Figure 5-14: Comparison between Automatic and Autonomous Mission Accomplishment (Framed Elements Form Work System).

In case A of Figure 5-14 the work system is consisting of a human operator (pilot) and the vehicle, the latter representing the work object and the operation-assisting means, i.e., the conventional setup of a manned vehicle. In this configuration the operation-assisting means will provide diverse automatic functions. Having conventional manned vehicles or aircraft, an external command and control unit works out a mission order as a representation of the desired mission objective and passes it to the operator, who accomplishes the mission. Such a work system acts autonomously and co-operatively, depending on the current situation, the goals, the system's and operator's capabilities and resources.

Case B of Figure 5-14 represents the solution of conventional automation to the guidance of an uninhabited vehicle (compare Figure 5-7). In this setup the work system is spatially dislocated, bearing the aforementioned restraining factors for remote operation. The vehicle itself may be considered as being semi-automatic in the case of loose supervision or even fully automatic if no monitoring or supervision is desired at all. Such a vehicle typically has no 'on-board intelligence', and therefore, will accomplish a mission automatically. In some occasions, if there is a person on ground acting as a remote operator within the guidance loop, he has some influence on the actions of the vehicle during operation and the vehicle acts partially automatically. Otherwise, the person takes more the role of a supervisor, who usually provides the vehicle with pre-planned instructions, possibly including some action alternatives. How adequate an automatic vehicle reacts to a situation change depends in case of operator-guided operation on whether the operator gets enough information about the situation in which the vehicle is located. If a vehicle operates fully automatically, it can only react to situation changes, which were foreseen by the operator.

Case C of Figure 5-14 is the situation where an autonomously performing work system is only consisting of machine elements, i.e., the vehicle including operation-assisting means and the ACU, which is capable of generating human-like behaviour. Exclusively in this configuration the remote agent (i.e., the vehicle and its

guidance) forms an autonomous entity itself. Having this capability on-board several vehicles with partially overlapping (i.e., partly equal and partly different) resources and capabilities, it becomes possible to have a mission accomplished autonomously and co-operatively with an external supervisor providing an overall mission objective to all of them. [10]

5.3.2.3 Cognitive Automation as Part of the Operation-Assisting Means

In a traditional sense the human operator provides capability of cognition within a conventional work system, whereas the operation-assisting means do not. As an alternative to full autonomy without human intervention a configuration, where an artificial cognitive component in addition to the human operator might be introduced into the work system.

Figure 5-15 [24] shows the ACU being part of the operation-assisting means in an otherwise conventional, manned work system setup.

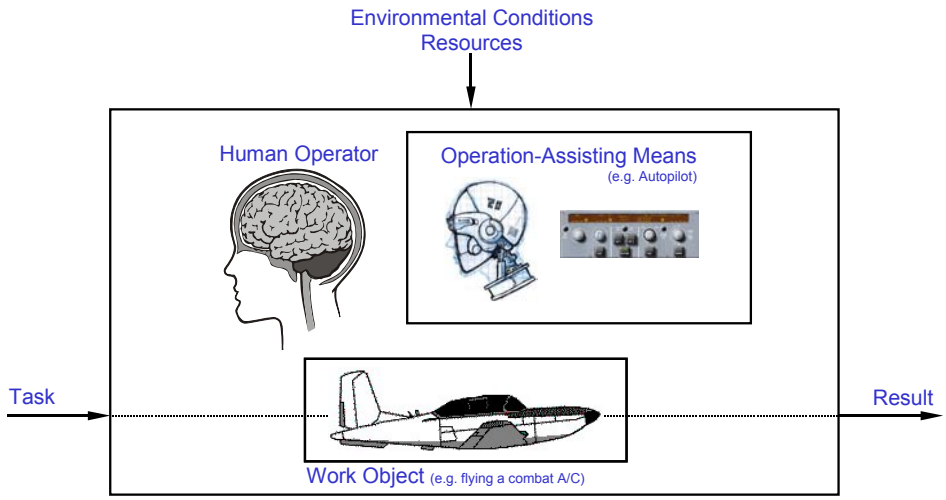


Figure 5-15: Work System with ACU in Configuration "Cognitive Automation as Part of Operation-Assisting Means".

> *"As opposed to conventional automation, cognitive automation works on the basis of comprehensive knowledge about the work process objectives and goals […], pertinent task options and necessary data describing the current situation in the work process. […] Making use of these capabilities in terms of operation-assisting means in the work system, it has no longer to be the exclusive task of the [human] operator to monitor the process subject to the prime work system objectives."* [24]

In the case of cognitive automation incorporated into the operation assisting means the vision of a "cognitive autopilot", as opposed to the conventional autopilot mentioned earlier on, would certainly perform superiorly. Once activated, a "cognitive altitude acquire" function would check the mode selection and the demand setting against the context of the current mission task. It would notice ground or traffic proximity, or maybe exposure to enemy radar. It would conclude that these conditions will result in loss of the mission or even disaster. Finally, it would work out an appropriate solution, either by indicating to the human operator the disturbance or by suggesting or even performing corrective actions.

This is pretty much the basic idea of a cognitive assistant system. Several research activities proved this concept more or less recently, the Cockpit Assistant System CASSY [11], the Crew Assistant Military Aircraft CAMA [12,13], and the Tactical Information and Mission Management System TIMMS [14]. Some more information on these projects will be given at the end of this chapter. Onken summarises the requirements for this class of systems:

> *"(1) It must be ensured the representation of the full picture of the flight situation, including that the attention of the cockpit crew is guided towards the objectively most urgent task or sub-task as demanded in that situation.*
>
> *(2) A situation with overcharge of the cockpit crew might come up even when situation awareness has been achieved by the pilot crew. In this case the assistant system has to transfer the situation into a normal one which can be handled by the crew in a normal manner."* [15]

In these so-called two basic requirements for human-machine interaction the way is paved already for the next step in the integration of cognitive automation in a work system, in the sense of cognitively facilitated human-machine co-operation as another alternative work system configuration.

5.3.2.4 Co-operative Automation as By-Product of Cognitive Automation

As opposed to mere interaction, co-operation has particular characteristics. Co-operating units in a work system pursue additional goals. Billings [2] formulates respective design principles for human-machine co-operation in the context of human centred design:

The human operator must be

- Actively involved;
- Adequately informed; and
- Able to monitor the automation assisting him.

The automated systems must

- Be predictable; and
- Also be enabled to monitor the human operator.

And,

- Every intelligent system element must know the intent of other intelligent system elements.

Billings, though, does not offer a solution how the intelligent machine elements shall be designed, yet. He does not bear machine-machine co-operation in mind, either.

Figure 5-16 [24] shows a work system setup, where the human operator and the ACU form a team. In this configuration the ACU has reached

> *"[...] the high-end authority level for decisions in the work system, which was, so far, occupied by the human operator alone."* [24]

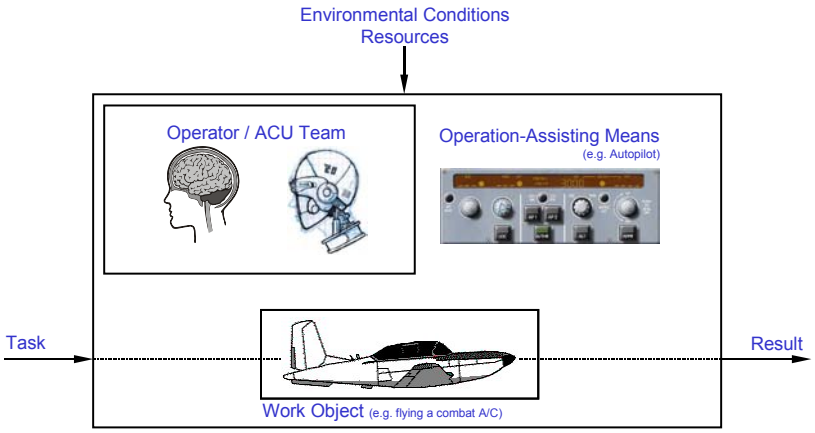


Figure 5-16: Work System with ACU in Configuration "Co-operative Automation".

As a consequence of this consideration each of the team members has to have the ability to carry out all tasks, which might be crucial for the performance of the overall work task. A crew co-ordination concept, very similar to one examined for human-human cockpit teams [3,16], has to be developed.

5.3.3 Technological Challenges

In the previous section the introduction of artificial cognitive capabilities in a work system was discussed. The perspective of this advanced automation technology approach is to overcome current problems with clumsy systems in human-machine co-operation, in order to facilitate machine autonomy without human intervention, and to support human operators in demanding tasks, which tend to overload human resources. The term of an Artificial Cognitive Unit (ACU) has been introduced, so far without explaining, how such a system element shall be constructed.

Figure 5-17 visualises the various technological challenges to be borne in order to implement such a system:

- Comprehensive situation perception: Figure 5-5 shows that in principle the human operator on-board has access to information (environmental stimuli) which is offered to him in addition to the information from his vehicle systems. The human operator is able to look out of the window of his vehicle; he can hear environmental noise or follow the voice communication on the radio. He can sense structural vibrations of his vehicle and even smell smoke in the cabin, however, the most important point is probably that the human operator has the principal capability to understand most of these perceptions and put them into the context of previous experiences. Conventional automation is lacking most of these abilities, and thereby, has no access to a wide spectrum of environmental information relevant for crucial decisions. In order to facilitate cognitive behaviour in a machine system the ability to perceive the environment has to be ensured. Dickmanns and his research group contributed very substantial work in the area of computer vision for autonomous road vehicle guidance (e.g., [17]).
- Cognitive capabilities: The next step after a successful perception of the world will be the deduction of rational behaviour on the basis of the gathered information. Therefore, further cognitive capabilities (of course, perception is a cognitive capability itself already) will be needed, both, on the human operator's side as well as on behalf of the machine. What humans can do seemingly effortlessly has to be given to the automation by design. Automation shall be enabled to built up a mental model of the surrounding world, which can be understood as the comprehension of the situation and its projection into the future (e.g., [18] as one point of view). The so-gained situational

knowledge shall be adequately represented in memory (e.g., [19] or [20] as two classical sources). On the basis of this situation specific knowledge and other pre-recorded knowledge, problem-solving and decision-making shall be performed in order to achieve certain goals. The modelling of this component of cognition will be the main subject of the next section of this chapter, resulting in the theory of the Cognitive Process [24].

- Human-machine interaction: Having an intelligent unit within the work system, which is enabled to gather and understand the entire situation, to make decisions and to exhibit rational and goal-oriented behaviour, it will be necessary to make it interact with the human operator. First of all, appropriate communication channels have to be found. A system designed to perform on the higher levels of cognition certainly offers the principal opportunity to use language as communication code [21], besides others. Furthermore, an appropriate co-ordination technique has to be found in order to facilitate a fruitful co-operation aiming upon the accomplishment of a common mission objective. In the long term, intelligent machines shall appreciate other intelligent agents in their environment, either human or artificial, as such. In this case, co-operation will be an additional behaviour of a machine, based upon cognitive capabilities [10].
- Level of automation and authority: Like with human teams, the question of the allocation of tasks and authorities has to be answered for human-machine teams, as well as for machine-machine teams. Weiner [3] investigated the issue of crew resource management for the aviation domain. Billings [2] made his suggestions for human centred aircraft automation design and human-machine co-operation. Taylor [22] worked on the problem of allocation of authorities within a human-machine team with the aim to provide the necessary and sufficient levels of authority for the task automation – but still, only the existence of artificial cognitive team mates will reveal the critical questions in the context of the allocation of tasks and authority, which have to be tackled.
- Paradigm shift: Finally, users, consumers, designers, companies, procurement officers, and customers, who are involved in the introduction of a new automation technology in their specific way, have to reconsider the issue of the evolution of personal, social, and economic factors, which comes along with such a process. In some cases a paradigm shift will be inevitable. At the very deep end of the chain, training issues related with the handling of somehow intelligent machinery will certainly come up.

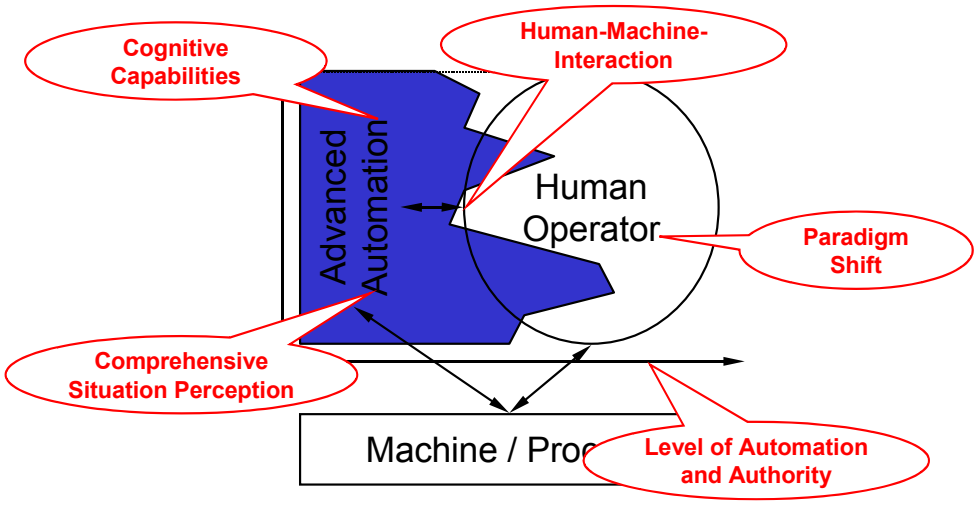


Figure 5-17: Technological Challenges in Advanced Automation.

The following section "Approaching Cognition" will be dedicated to the analysis of cognitive capabilities of humans. On the basis of this, an overview over information technology approaches to artificial cognition will be given. As a result, the so-called Cognitive Process (CP) as a theoretical approach to machine intelligence will be introduced.

5.3.4 References

[1] Onken, R. (1999). The Cockpit Assistant Systems CASSY/CAMA. In: World Aviation Congress 1999. 99WAC-91. San Francisco, October.

[2] Billings, C.E. (1997). Aviation Automation: The Search for a Human-Centered Approach. Lawrence Erlbaum.

[3] Wiener, E.L. (1989). Human factors of advanced technology ("glass cockpit") transport aircraft (Technical Report 117528). Moffett Field, CA: NASA-Ames Research Center.

[4] Wiener, E.L. and Curry, R.E. (1980). Flight Deck Automation: Promises and Problems. In: Ergonomics. 23(10):995-1011.

[5] Wiener, E.L. and Nagel, D.C. (1987). Human Factors in Aviation. Academic Press. 1988.

[6] Bainbridge, L. (1987). Ironies of Automation. In: New Technology and Human Error. Eds.: Rasmussen, Duncan, Leplat. Wiley.

[7] Rasmussen, J., Pejtersen, A.M. and Goodstein, L.P. (1994). Cognitive Systems Engineering. Wiley.

[8] Anderson, J.R. (2000). Cognitive Psychology and its Implications. Fifth Edition. Worth Publishers.

[9] Eysenck, M.W. (1993). Principles of Cognitive Psychology. Lawrence Erlbaum Associates Publishers.

[10] Ertl, C. and Schulte, A. (2004, September). System Design Concepts for Co-operative and Autonomous Mission Accomplishment of UAVs. In: Deutscher Luft- und Raumfahrtkongress. Dresden, GE. 20-23.

[11] Prévôt, T., Gerlach, M., Ruckdeschel, W., Wittig, T. and Onken, R. (1995). Evaluation of intelligent on-board pilot assistance in in-flight field trials. In: 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems. Massachusetts Institute of Technology, Cambridge, MA. June.

[12] Walsdorf, A., Onken, R., Eibl, H., Helmke, H., Suikat, R. and Schulte, A. (1997). The Crew Assistant Military Aircraft (CAMA). In: The Human-Electronic Crew: The Right Stuff? 4th Joint GAF/RAF/USAF Workshop on Human-Computer Teamwork. Kreuth, GE. September.

[13] Schulte, A. and Stütz, P. (1998). Evaluation of the Crew Assistant Military Aircraft (CAMA) in Simulator Trials. In: NATO Research and Technology Agency, System Concepts and Integration Panel. Symposium on Sensor Data Fusion and Integration of Human Element. Ottawa, Canada. September 14-17.

[14] Schulte, A. (2002). Cognitive Automation for Attack Aircraft: Concept and Prototype Evaluation in Flight Simulator Trials. In: International Journal of Cognition Technology and Work. MS No. 94, Vol. 4 No. 3, Pages 146-159. Springer London Ltd.

[15] Onken, R. (1994). Basic Requirements Concerning Man-Machine Interactions in Combat Aircraft. Workshop on Human Factors/Future Combat Aircraft. Ottobrunn, Germany. October.

[16] Wiener, E.L. (1993). Intervention strategies for the management of human error. (NASA Contractor Rep. No. 4547) NASA Ames Research Center, Moffett Field, CA.

[17] Dickmanns, E.D. (2002). Expectation-based, multi-focal, saccadic (EMS) vision for ground vehicle guidance. In: Control Engineering Practice 10 (2002) , pp. 907-915. Pergamon Elsevier Science.

[18] Endsley, M.R. (2000). Theoretical underpinnings of situation awareness: A critical review. In: Endsley & Garland (Eds). Situation Awareness Analysis and Measurement. Lawrence Erlbaum Associates.

[19] Quillian, M.R. (1966). Semantic Memory. Bolt, Beranak and Newman. Cambridge, MA.

[20] Minsky, M. (1974). A Framework for Representing Knowledge. Memo 306. Cambridge, MA. MIT AI Lab.

[21] Gerlach, M. and Onken, R. (1993). Speech Input/Output as Interface Device for Communication between Aircraft Pilots and the Pilot Assistant System CASSY. In: Applications of Speech Technology. Joint ESCA-NATO/RSG 10 Tutorial and Workshop. Lautrach, GE.

[22] Taylor, R.M. (2001). Cognitive Cockpit Engineering: Pilot Authorisation and Control Tasks. In: 8th Conference on Cognitive Science Approaches to Process Control (CSAPC). Munich.

[23] Schulte, A. (2003). Systems Engineering Framework Defining Required Functions of Un-inhabited Intelligent Vehicle Guidance. In: NATO RTO. Human Factors and Medicine Panel. Task Group HFM-078 on Unmanned Military Vehicles: Human Factors in Augmenting the Force. Leiden, NL. 10th – 13th June.

[24] Onken, R. (2002). Cognitive Cooperation for the Sake of the Human-Machine Team Effectiveness. In: RTO-HFM Symposium on The Role of Humans in Intelligent and Automated Systems. Warsaw, Poland. 7-9 October.

[25] REFA (1984). (Verband für Arbeitsstudien und Betriebsorganisation e.V.). Methodenlehre des Arbeitsstudiums. Teil 1: Grundlagen. Hanser-Verlag.

[26] Sheridan, T.B. (1992). Telerobotics, Automation and Human Supervisory Control. MIT Press.

5.4 APPROACHING COGNITION

In the previous sections the term “cognition” has been used rather sloppy in the sense of a particular human capability, and, hopefully, of a future machine function. This section shall sort things out in terms of how humans perform and how a machine has to be constructed in order to exhibit intelligent behaviour, likewise. “Intelligence”, a term which can be replaced by “cognition” in most cases in this context, is defined rather

vaguely defined in habitual language use, although being a rather valid concept in psychology. Besides many other definitions, Morris [1] gives:

> *"Intelligence [… is …] a general term encompassing various mental abilities, including the ability to remember and use what one has learned, in order to solve problems, adapt to new situations, and understand and manipulate one's environment."* [1]

Nowadays intelligence or cognition is no longer exclusively considered by psychology, but is subject to the interdisciplinary field of "cognitive science", which is influenced by philosophy, psychology, neuroscience, linguistics, anthropology, and, of course, by computer science and information technology. In this enumeration the last discipline seems to be of some particular interest, because it facilitates to prove the validity of theories by modelling and simulation. New concepts emerging in the field of Artificial Intelligence (AI), a field of computer science that attempts to develop intelligently behaving machines [Anderson, 2000], influenced the cognitive psychology, and vice versa [2].

Many approaches dedicate themselves to the exploration of the underlying processing structure, as opposed to the principles of behaviourism, which was a rather strong trend in the early 20th century psychology, only being concerned with the externally observable behaviour of a human.

Figure 5-18 depicts the different approaches, the one of the behaviourism (top), and the alternative modelling view considering the internal processing (bottom). In both cases, the information processing paradigm (input → processing → output) is appropriate to characterise the phenotype of the situation. The behaviourism searches for the input-output mapping of human behaviour, no matter how it will be implemented. Other modelling approaches focus on the description of the underlying processes, in order to expose the observed behaviour. In the subsequent few sections a brief overview will be given over approaches to the modelling of the processing mechanisms of human cognition. Behaviour, in turn, can be utilised in order to validate related models.

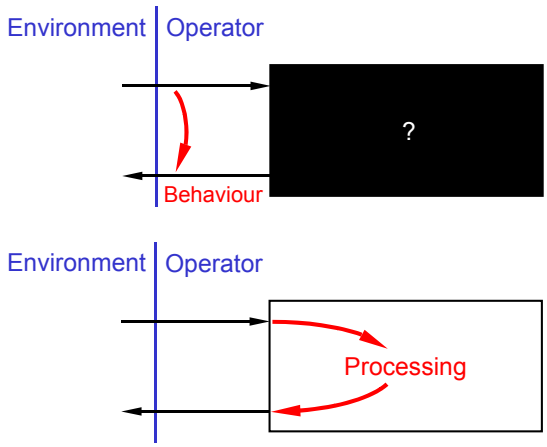


Figure 5-18: Modelling Behaviour or Processing.

5.4.1 Model of Human Performance

In order to open the window to the development of human-like performance features in terms of cognitive automation, the very well accepted model of human performance levels by Jens Rasmussen [3] will be

consulted, to begin with. The simplicity and intelligibility made this model, originally having its seeds in ergonomics research, quite popular in the circles of cognitive psychologists as well as amongst engineers. In fact, Rasmussen's model became the probably most common psychological scheme within the entire engineering community.

Without going into too much detail here (for a most detailed discussion refer to [3] and [26]), the model distinguishes between three levels of human performance, the skill-based, the rule-based, and the knowledge-based behaviour (see Figure 5-19). On the skill-based level highly automated control tasks will be performed, without any mental effort or consciousness. Typical for this level is the continuous control of the body in three-dimensional space and time. Most of this performance is carried out in feedforward control mode by pre-programming of stored sensor-motor patterns on the basis of task specific features. Typical behaviour on this level, like tracking a road, will be assembled by running a sequence of parameterised templates with some feedback control ratio for precision enhancement.

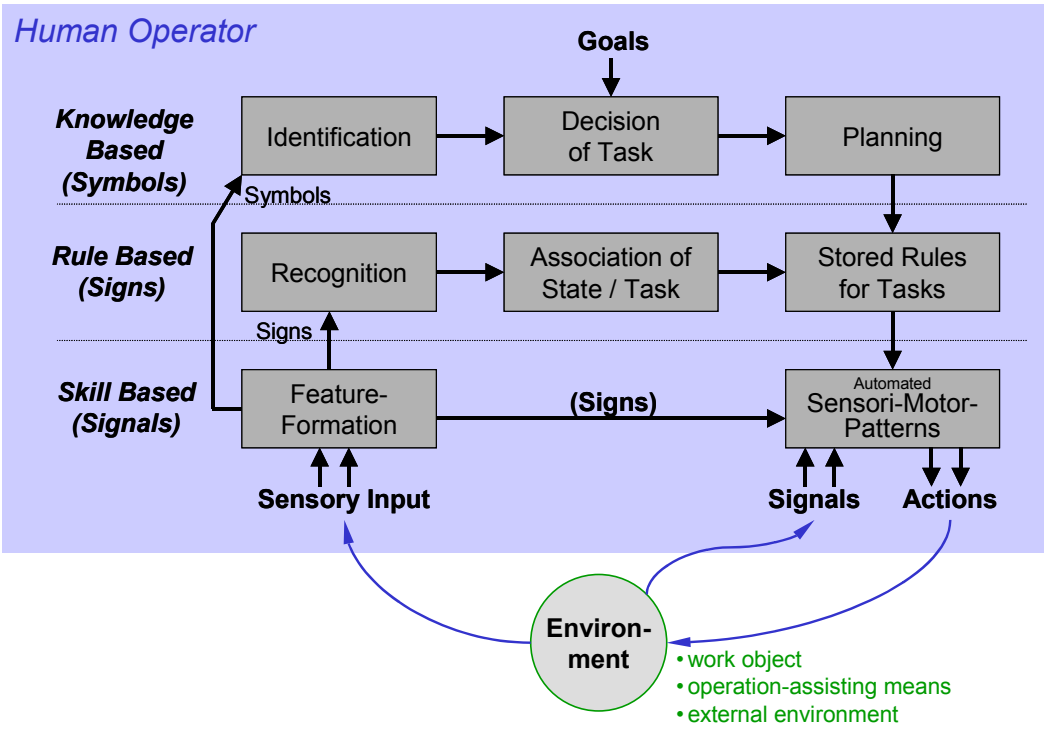


Figure 5-19: Rasmussen's Model of Human Operator's Performance Levels Linked to Environment.

On the rule-based level most of the everyday conscious action that we perform takes place in a strict feedforward control manner. Here, humans follow pre-recorded scripts and procedures in order to activate the appropriate sensori-motor patterns on the basis of the presence of clearly recognised objects characterising the prevailing situation. With training formerly rule-based performance tends to be dropped to the skill-based level. Rule-based performance is goal-oriented, although goals are not explicit, but encoded in the pre-conditions of the applicable rules.

The knowledge-based level will be entered in situations, where there are no applicable rules available in order to recognise objects or to determine the selection of action. This is the case when the situation requires the

preoccupation with a non-pre-defined problem. In this case general concepts have to be consulted in order to identify the situation, i.e., find similar or somehow related situations in previous experience. Goals derived from overall aims explicitly direct the tasking. Planning, i.e., problem-solving will be deployed in order to generate new scripts or procedures, which will be executed on the rule-based level. In general, problem-solving can be considered as a highly versatile process, incorporating strategies such as difference reduction and means-ends analysis [17] as well as search in problem space [4]. So-called mental models will be the knowledge basis for the highest performance level [3].

Although it is so common to the engineering community, because of its apparent use of clear functional blocks and their interrelations, Rasmussen's model deserves some interpretation from an information technology point of view. One reason for this is the improper handling of knowledge in the model. Most of the boxes represent a dedicated function or processing step (e.g., 'recognition', 'planning'). Only two particular boxes (i.e., 'stored rules for tasks' and 'sensori-motor patterns') represent knowledge, without having their individual functions specified. And finally, only one functional block (i.e., 'decision of task') makes use of an explicit knowledge basis ('goals'). From an information technology standpoint it would be desirable to modify the model according to the following guidelines, at least for a first step of advancement:

- Use boxes for functions or processing steps;
- Label the knowledge which is made use of in each box; and
- Label all inputs and outputs of the functional blocks.

The detailed discussion of this issue shall be the matter of forthcoming publications.

5.4.2 Modelling Approaches for Intelligent Machine Behaviour

As discussed above, the human performance can be decomposed in several high level cognitive functions, which rely upon certain a-priori knowledge. Besides the task-related a-priori knowledge, there are mechanisms necessary in order to process this knowledge. Highly related with these mechanisms is the form of representation of this knowledge. In parallel to the development of psychological performance and behaviour models as briefly discussed in the previous section there takes place the development of technological approaches to intelligent machine behaviour, each of which influencing and fertilising one another.

From a very global standpoint there can be identified two fundamentally different approaches, one strongly influenced by the idea of mimicking the human implementation of cognition in the brain (i.e., connectionism, artificial neural networks, sub-symbolic AI) [5], and the other being based upon models taken from information technology (i.e., symbolism, Artificial Intelligence) [2].

Besides those two main streams, early human factors research offered modelling approaches on the basis of control theory [18].

Figure 5-20 shows the principal approach of this class of approaches, modelling human behaviour by means of transfer functions. Typically, there were made a couple of structural assumptions, such as reaction time and neuromotor delay as inherent parameters and gain and anticipation as task-adaptable parameters. On the basis of such model structure quite successful parameter identifications could be performed, typically limited to various sensori-motor control tasks.

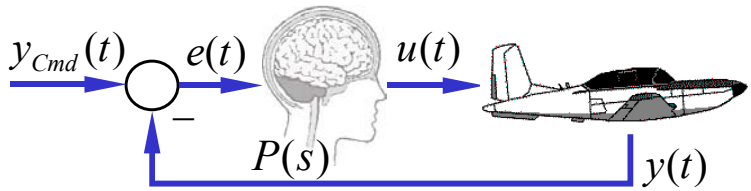


Figure 5-20: Model of Human Behaviour Motivated by Control Theory (i.e., Transfer Function).

Coming back to the aforementioned antithetic approaches of connectionism and symbolism one major difference can be identified in the way of knowledge representation. In the connectionism there is no separation existing between knowledge and its processing. Neither is knowledge in any way explicit, but spread over the weights of the connections between simple but numerous processing units (neurons). Each single weight provides a contribution to the knowledge persistent to the model without a particular allocation of meaning. The entirety of weights represents the entirety of a-priory knowledge. Many models provide learning mechanisms, either in supervised or unsupervised learning mode.

Symbolism, on the other hand, utilises explicit, meaningful symbols in order to handle knowledge. Processing architectures are derived from simple information processing paradigms, as depicted in Figure 5-21.

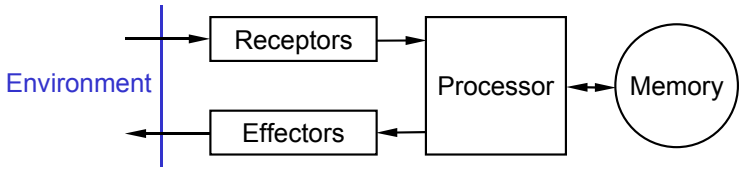


Figure 5-21: Model of Human Processing Motivated by Information Technology.

While the processor is almost independent from the task, the functionality is encoded in the knowledge persistent to the memory. The interface to the external world build dedicated receptors and effectors. The probably most famous, classical model of this kind is the so-called CMN-model [6] as shown in Figure 5-22.

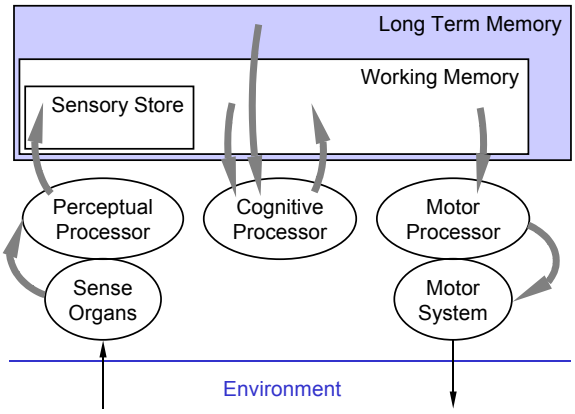


Figure 5-22: The Model Human Processor Adapted from CMN-Model.

The CMN-model in particular points out the assumed structure of the memory of the human and some performance features and limitations of its building blocks. The 7-chunk capacity limit of the working memory is probably one of the most acquainted proposition in this context. As principal concept of processing the so-called recognise-act cycle (RAC) (see Figure 5-23) is proposed.

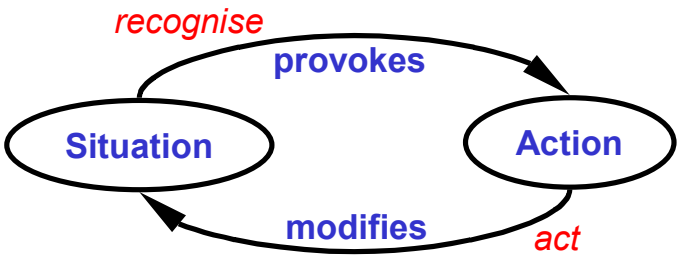


Figure 5-23: The Recognise-Act Cycle.

To characterise the activity of the cognitive processor, [6] state:

> *On each cycle, the contents of Working Memory initiate associatively-linked actions in the Long-Term Memory ("recognize"), which in turn modify the contents of Working Memory ("act"), setting the stage of the next cycle.* [6]

The interface to the environment is through the working memory.

The so-called production systems (expert systems) predominantly follow the processing approach of the recognise-act cycle using mainly IF-THEN rules as knowledge representation form for heuristics and "rules of thumb". Figure 5-24 shows the main building blocks of such a rule-based system (i.e., production system). The knowledge is stored in the rule base, the long-term memory of the architecture. Based upon the short-term (i.e., working) memory contents (i.e., internal states plus input from and output to the environment) according to their pre-conditions rules from the rule base will be selected as candidates for execution. After the solution of conflicts (in the case of, e.g., more than on applicable rules) the rule will be "fired", i.e., the post-condition of the rule will be executed in order to modify the content of the short-term memory, either initiating a succeeding recognise-act cycle base on internal state changes, or evoking an action at the output.

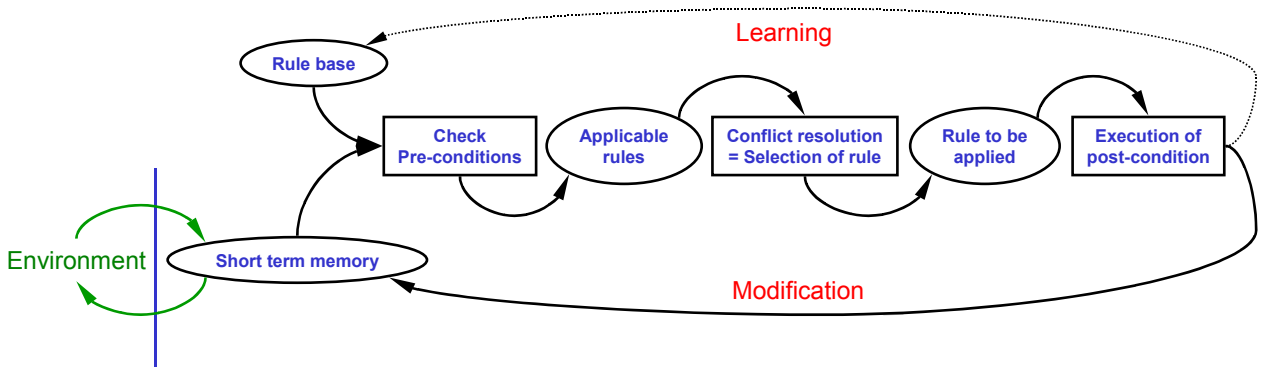


Figure 5-24: Architecture of a Rule-Based System (i.e., Production System).

Besides these very traditional approaches, predominantly relying on the use of rules as form of knowledge representation, many other kinds of knowledge representations evolved in the era of GOFAI ("Good-Old-Fashioned Artificial Intelligence"), most of which linked to symbolist approaches on one or the other way, e.g., semantic networks [19], conceptual dependency [7], frames/schemata [20], scripts [8], just to name the classical ones.

Besides these "classical" ones there are at least two more recent approaches important to be mentioned here, both of which being symbolic cognitive architectures meant to model intelligent performance:

- ACT-R [9] is used to model different aspects of human cognitive behaviour, i.e., to implement human-like behaviour. ACT-R has its starting point in creating a computational theory of human memory. It combines predominantly symbolic representations with sub-symbolic mechanisms, mainly to model human performance aspects such as the limited retrievability of knowledge.
- SOAR [10,4] is used to model an agent's intelligent capabilities, i.e., to implement rational behaviour. SOAR has its roots in the attempt to understand the methodological and structural pre-requisites of human problem-solving and decision-making. Concerning knowledge representations, SOAR is a rule-based, i.e., a production system.

While the aforementioned architectures pair a still strong focus on knowledge representation with architectural aspects of cognition, some concurrent approaches capitalise upon mostly architectural views. Some of the most prominent approaches shall be brought up here:

- BDI (Belief-Desire-Intent)-Agents [11]: Agents are software constructs situated in a certain environment and interacting with it autonomously in order to achieve specific individual objectives. Applications are widely spread over various domains from data management over user interfaces and computer mediated collaboration to robotics. The BDI architecture suggests the usage of mental attitudes representing the informational (belief), the motivational (desire) and the deliberative (intent) state of the agent.
- RCS (Real-time Control System) [12]: RCS is a reference model architecture, suitable for real-time control problem domains, and therefore closely related to robotics. It focuses on intelligent control that adapts to uncertain and unstructured operating environments. The architecture provides a top-down hierarchical composition of processing nodes incorporating the cognitive functions of sensory processing, world modelling, value judgement and behaviour generation.
- Subsumption Architecture [13], representing the field of behaviour-based robotics, almost fully dismisses the notion of a mental world model. Instead, this architecture is strongly behaviour oriented, i.e., focussing on direct perception-action mappings facilitated by close couplings between sensors and actuators. More complex behaviours are assumed to emerge from simpler ones in a bottom-up manner. Symbolic representations are not part of this architecture.

As this very brief, and by no means complete, overview of modelling approaches for intelligent machine behaviour indicates, the research focus over the last three decades has been shifted from mostly method oriented approaches, e.g., how to represent knowledge, to somewhat more architecture focussed approaches.

When it shall come down to a systems engineering implementation of intelligent machinery, both aspects yet are of their particular importance, and therefore should be considered in a well balanced manner. The following sub-sections introduce the concept of the Cognitive Process [21,22,14], which comprises a theory based on cognitive psychology with a knowledge-based architecture.

5.4.3 The Cognitive Process as Approach to Cognitive Automation

Coming back to the aim of a co-operative structure of a human-machine system (as depicted in Figure 5-12), the notion of cognitive automation (as introduced in the Sections 5.3.2. ff.), and the technological challenge of providing cognitive capabilities to an Artificial Cognitive Unit (ACU) (as formulated in Section 5.3.3), we now want to take the findings on cognition (Section 5.4) into consideration in order to develop a theory-based architecture for intelligent machine behaviour. Findings from cognitive psychology and artificial intelligence shall be taken into consideration likewise.

The concept of a piece of automation being a team-player in a mixed human-machine team, or even a machine taking over responsibility for work objectives to a large extent, promotes the approach of deriving required machine functions from models of human performance. In Section 5.4.1 Rasmussen's model has been introduced.

When we look at conventional automation as discussed in the Section 5.2.2 and 5.3.1, in particular in the avionics domain, it mainly acts on a level which might be compared with the skill-based human performance level (e.g., flight control systems, autopilot systems). Some functionalities might be attributed to the rule-based (e.g., traffic collision avoidance systems) and few on the knowledge-based level (e.g., mission planning support in flight management systems).

On the other hand, not many automation systems can be identified, providing an understanding of the current situation in terms of recognition and identification, or considering goals, which are essential for the decision of what to do next in an unknown situation, as already discussed in Section 5.3.1.

In contrast to the conventional approach, cognitive automation aims for rationality in a human-like performance manner, without modelling typical human's shortcomings. Thus, all functions of Rasmussen's model have to be covered, including those already incorporated in conventional automation [14]. Figure 5-25 shows the main focus area for future developments aiming at cognitive automation, namely the implementation of a comprehensive situation understanding and goal-driven decision-making, as high level cognitive capabilities.

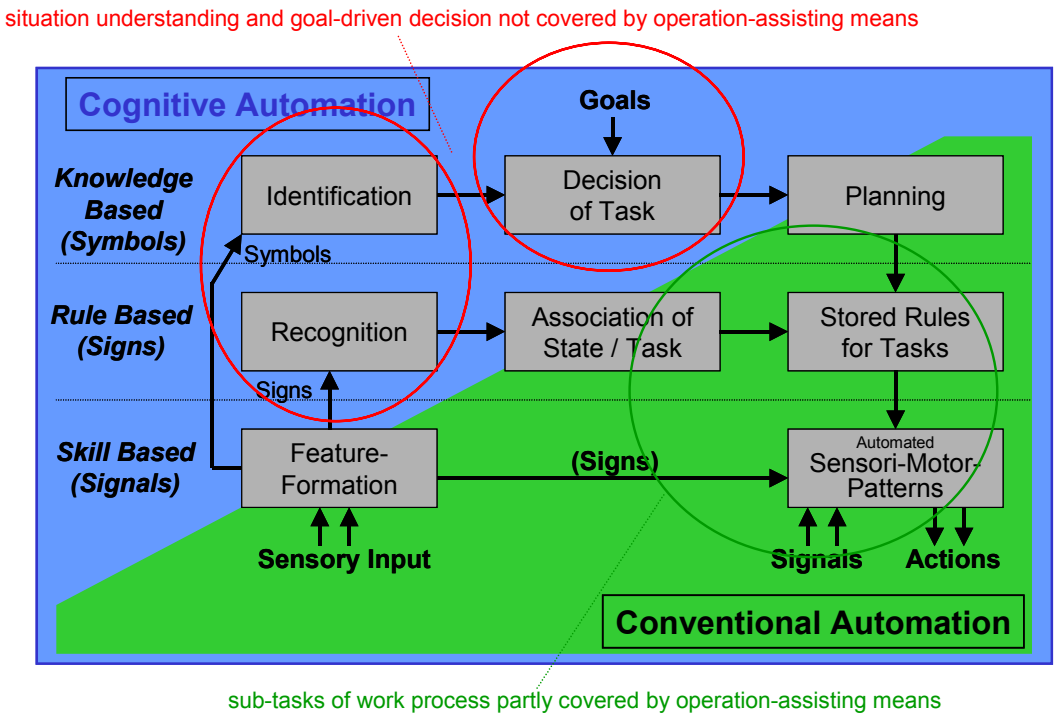


Figure 5-25: Conventional and Cognitive Automation Explained by Rasmussen's Model of Human Performance.

In order to achieve a system engineering framework, the main idea of Rasmussen's model, namely rule- and knowledge-based performance, is mapped into the so-called Cognitive Process (CP). The CP is an approach to modelling human information processing, which is suitable for providing human-like rationality [14,15]. As it is compatible with human cognition, and the generated behaviour is driven by goals, which are represented explicitly, it is well suited for the development of a cognitive system, which is part of a team consisting of artificial and/or human team mates.

Figure 5-26 shows the CP consisting of the body (inner part) and the transformers (outer extremities). The body contains all knowledge, which is available for the CP to generate behaviour. There are two kinds of knowledge: the 'a-priori knowledge', which is given to the CP by the developer of an application during the design process and which specifies the behaviour of the CP, and the 'situational knowledge', which is created at run time by the CP itself by using information from the environment and the a-priori knowledge. The functional units effectively processing knowledge are the above-mentioned transformers, which read input data in mainly one area of the situational knowledge, use a-priori knowledge to process the input data, and write output data to a designated area of the situational knowledge.

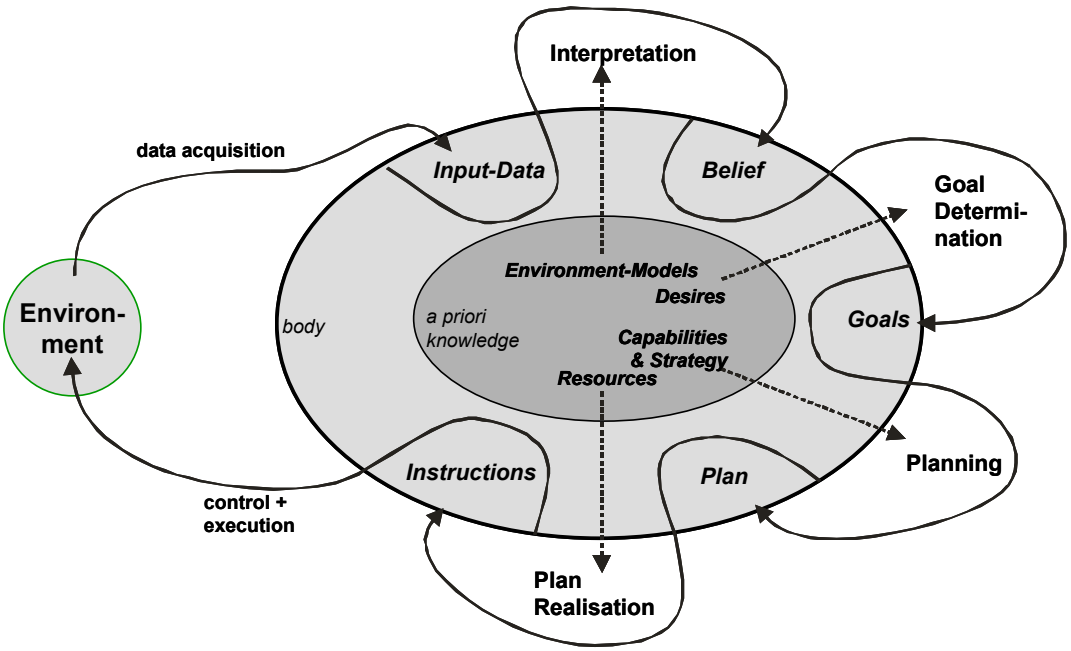


Figure 5-26: The Cognitive Process.

The following steps are performed by the CP in order to generate behaviour:

- Information about the current state of the environment (input data) is acquired via the input interface. In this context, the environment includes other objects in the physical world, e.g., another UAV or an obstacle, as well as the underlying vehicle of the CP. Therefore, the input data may for instance contain information about the current autopilot mode or pre-processed sensor information.
- The input data are interpreted to obtain an understanding of the external world (belief). The interpretation uses environment models, which are concepts of elements and relations that might be part of the environment, to build this internal representation.
- Based on the belief, it is determined, which of the desires (potential goals) are to be pursued in the current situation. These abstract desires are instantiated to active goals describing the state of the environment, which the CP intends to achieve.
- Planning determines the steps, i.e., situation changes, which are necessary to alter the current state of the environment in a way that the desired state is achieved. For this planning step, models of action alternatives of the CP are used.
- Instruction models are then needed to schedule the steps required to execute the plan, resulting in instructions.
- These instructions are finally put into effect by the appropriate effectors of the host vehicle. The resulting actions affect the environment, i.e., modify the physical world.

These functional units represent an application-independent inference mechanism, which processes application-specific knowledge. This knowledge-based design approach is of great advantage when implementing the CP: The inference mechanism has to be implemented only once, and can then be used for different applications.

It is desirable to reuse not only the inference mechanism, but also knowledge in different applications. For this purpose, a-priori knowledge has to be unitised in so-called 'packages', each of which represents a certain capability. As indicated in Figure 5-27, each package (depicted as horizontal layer) implements a capability which is designed according to the blueprint of the CP. Several packages together form the complete system. They are linked by dedicated joints in the a-priori knowledge and by the use of common situational knowledge. When looking vertically on the packages, a uniform structure of the a-priori knowledge and its order of usage in terms of processing steps according to the transformers of the CP can be recognised.

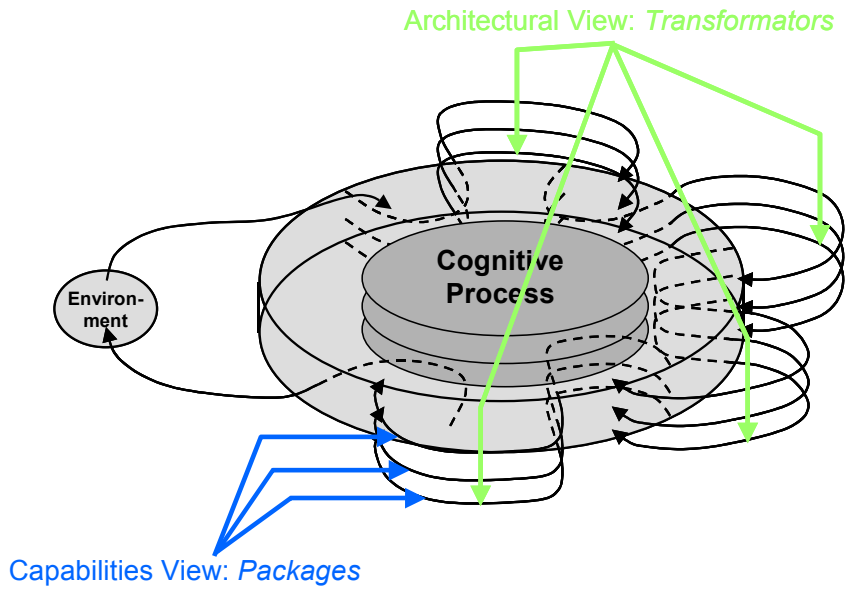


Figure 5-27: Representing Multiple Capabilities on Basis of the Cognitive Process.

5.4.4 References

[1] Morris, C. (Ed.) (1996). Academic Press Dictionary of Science and Technology. Academic Press.

[2] Newell, A. and Simon, H. (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice Hall.

[3] Rasmussen, J. (1983). Skills, rules and knowledge, signal, signs and symbols, and other distinctions in human performance models. In: IEEE Transactions on Systems, Man and Cybernetics. SMC-13, pp. 257-266.

[4] Newell, A. (1990). Unified Theories of Cognition. Harvard University Press. Cambridge, MA.

[5] Rumelhart, D.E., McClelland, J.L. and the PDP Research Group (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volumes 1 and 2. Cambridge, MA: MIT Press.

[6] Card, S.K., Moran, T.P. and Newell, A. (1983). The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, Publishers. New Jersey.

[7] Schank, R.C. (1975). Conceptual Information Processing. Elsevier. New York.

[8] Schank, R.C. and Abelson, R. (1977). Scripts, Plans, Goals, and Understanding. Erlbaum Associates. Hillsdale, NJ.

[9] Anderson, J.R. (1993). Rules of Mind. Erlbaum. Hillsdale, NJ.

[10] Laird, J., Newell, A. and Rosenbloom, P. (1987). SOAR: An Architecture for General Intelligence. Artificial Intelligence, 33.

[11] Rao, A. and Georgeff, M. (1995). BDI Agents from Theory to Practice. Technical Note 56. AAII. April.

[12] Albus, J.S. and Meystel, A.M. (2001). Engineering of Mind: An Introduction to the Science of Intelligent Systems. John Wiley & Sons, Inc.

[13] Brooks, R.A. (1991). How to build complete creatures rather than isolated cognitive simulators. In: K. Van Lehn (ed.). Architectures for Intelligence. Lawrence Erlbaum Associates. Hillsdale, NJ.

[14] Putzer, H. and Onken, R. (2003). COSA – A Generic Cognitive System Architecture based on a Cognitive Model of Human Behavior. In: International Journal of Cognition Technology and Work.

[15] Ertl, C and Schulte, A. (2005). Enabling Autonomous UAV Co-operation by Onboard Artificial Cognition. In: Proceedings of 1st International Conference on Augmented Cognition, in conjunction with HCI International, Las Vegas, USA, 22nd – 27th July.

[16] Rasmussen, J., Pejtersen, A.M. and. Goodstein, L.P. (1994). Cognitive Systems Engineering. Wiley.

[17] Anderson, J.R. (2000). Cognitive Psychology and its Implications. Fifth Edition. Worth Publishers.

[18] Rouse, W.B. (1980). Systems Engineering Models of Human-Machine Interaction. Elsevier North Holland.

[19] Quillian, M.R. (1966). Semantic Memory. Bolt, Beranak and Newman. Cambridge, MA.

[20] Minsky, M. (1974). A Framework for Representing Knowledge. Memo 306. Cambridge, MA. MIT AI Lab.

[21] Onken, R. and Walsdorf, A. (2000). Assistant Systems for Vehicle Guidance: Cognitive Man-Machine Cooperation. In: 4th International Conference on IT for Balanced Automation Systems – BASYS 2000. Berlin, Germany. 27-29 September.

[22] Onken, R. (2002). Cognitive Cooperation for the Sake of the Human-Machine Team Effectiveness. In: RTO-HFM Symposium on The Role of Humans in Intelligent and Automated Systems. Warsaw, Poland. 7-9 October.

5.5 COGNITIVE SYSTEMS ARCHITECTURE – REALISATION ASPECTS

This section is supposed to point out a perspective of how to implement an Artificial Cognitive unit (ACU) on the basis of the proposed theory. Figure 5-28 depicts what has been achieved so far, as a review of the previous sections. The starting point is the human operator as operating element in a work system. In a first

step we model human performance in terms of high level cognitive functions. The analysis of the typical work share in work systems reveals that there are particular shortcomings in terms of these high level cognitive functions on the machine side, namely in the domain of situation understanding and goal-driven behaviour. In order to achieve this capability on the machine side the Cognitive Process is proposed as underlying theory derived from useful findings in cognitive psychology and artificial intelligence research, likewise.

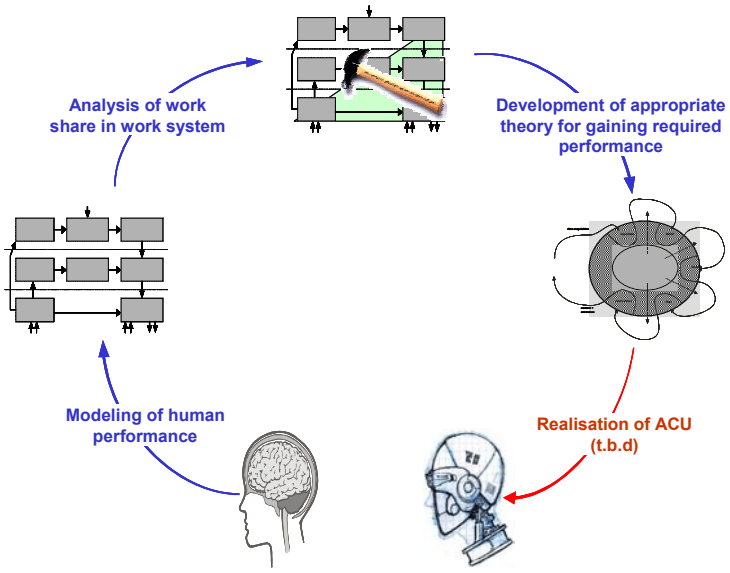


Figure 5-28: Method of Cognitive Systems' Development.

As a next step of the development of a cognitive system according to the approach of cognitive automation, the realisation of an Artificial Cognitive Unit (ACU) has to be accomplished. In the succession of this section an engineering framework for the development of such an ACU will be described: The Cognitive System Architecture (COSA).

COSA offers a framework to implement applications according to the theory of the Cognitive Process. It provides an inference mechanism and various means, which make it possible for the developer of an application to use concepts like 'belief', 'goal' and 'plan' rather than a programming paradigm based on a functional decomposition of an application [1].

COSA is composed of four building blocks (cf. Figure 5-29).

- The kernel implements the theory of the CP and does not contain any application-dependent information. Its only task is to generate behaviour from knowledge. In the current implementation, it is based on SOAR [Laird et al., 1987], which is a general rule-based architecture for developing systems that exhibit intelligent behaviour, as described earlier in this chapter. The CP-Library structures the situational knowledge according to the theory of the Cognitive Process and coordinates the performance of the transformers (interpretation, goal determination, planning, and plan realisation, cf. Figure 5-26).
- The application is formed by several COSA-compliant application components, which correspond to packages (see Figure 5-27). The components provide the a-priori knowledge and may also contain servers with interfaces to the environment or for external calculations.

- The front end provides tools for the developer of an application, which help him to model the knowledge for the application. One part of the front end is CPL (Cognitive Programming Language), which provides a programming support to the implementation of the above-mentioned mental notions 'belief', 'goal', and 'plan' as concepts for knowledge modelling. The CPL code, which represents the knowledge on a rather high abstraction level, is compiled into a code representation understood by the kernel, i.e., SOAR in the current implementation.
- Finally, the distribution layer is responsible for the communication among the modules of COSA. It ensures that components and modules can run on different computers in a network.

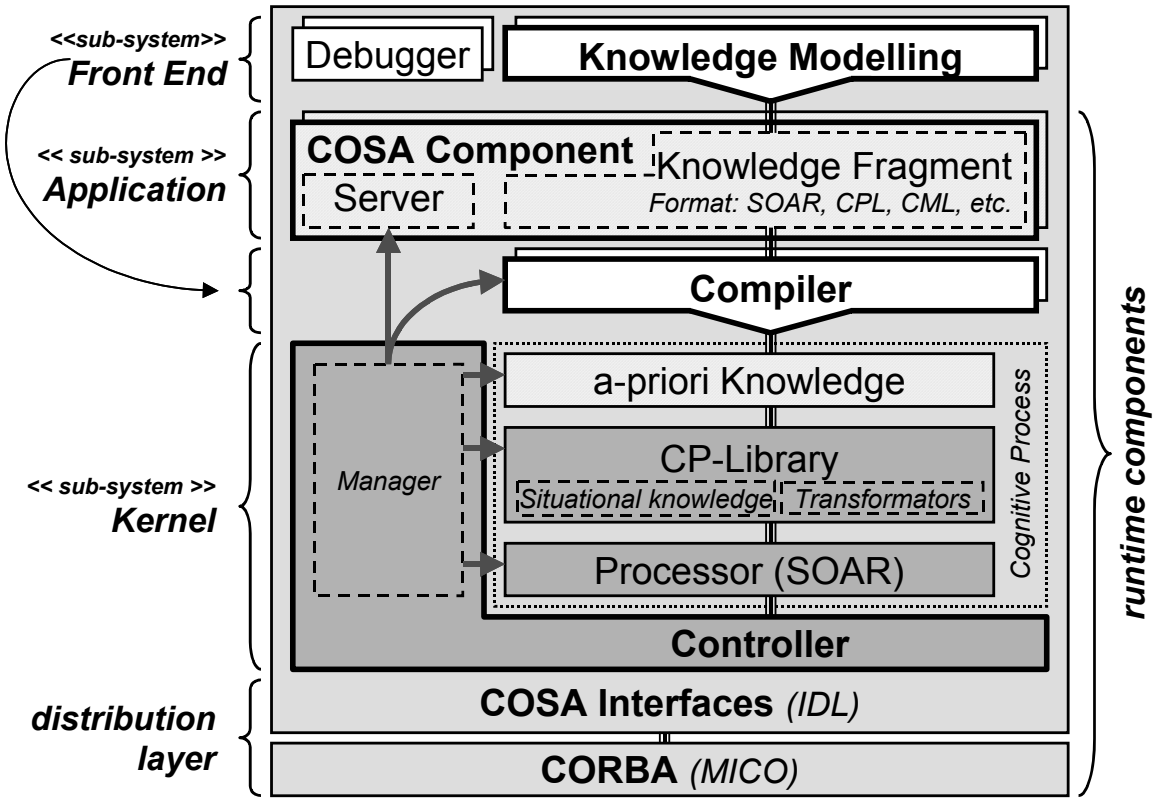


Figure 5-29: COSA – Cognitive System Architecture.

5.5.1 References

[1] Putzer, H. and Onken, R. (2003). COSA – A Generic Cognitive System Architecture based on a Cognitive Model of Human Behavior. In: International Journal of Cognition Technology and Work.

5.6 APPLIED SYSTEM APPROACHES

The previous sections provided a rather general concept of how to approach artificial cognition and co-operative automation, which represents the current state of the art achieved at the Institute of System Dynamics and Flight Mechanics at the Munich University of the German Armed Forces, at least from a theoretical standpoint. On the practical side this achievement could be attained during the course of several

experimental programmes in the field over the recent 15 years. Most of the works were well documented by multiple publications and internationally well recognised, e.g., [6] reporting the results of the flight trials of the worlds first comprehensive knowledge-based assistant system for flight-deck crews, CASSY (Cockpit Assistant System). In the late nineties followed CAMA (Crew Assistant Military Aircraft) as a prototype system in the military transport domain, e.g., [7] discussing the system architecture, [Schulte & Stütz, 1998] reporting on the successful simulator validation of the system, and [46] pointing out the successful flight tests that followed.

A detailed discussion of the achievements of the working group around Reiner Onken would go beyond the scope of this chapter. Instead of doing that, the rest of this chapter shall be dedicated to a broadening of the view by bringing up three perspectives from other research groups. The first contribution (Sub-section 5.6.1) is provided by Mike Chamberlin (UK) dealing with the issue of Artificial Intelligence from a methods' point of view. This treatment perfectly supplements the findings of Section 5.4, especially of Sub-section 5.4.2, which were pretty much focussed on a unitary approach. Sub-section 5.6.1 will give a nice overview and evaluation of further approaches.

Common to all attempts to computational intelligence of whatsoever kind is the (open) question to the acquisition of the necessary knowledge and it's adequate representation. In Sub-section 5.6.2 Jack Edwards (Canada) tackled the problem of intelligent, adaptive help system design mainly under the consideration a knowledge design and engineering methodology that combines elements of the CommonKADS and IDEF methods, Explicit Models Design and Perceptual Control Theory.

Finally, Sub-section 5.6.3 by Mike Waters and Robert Taylor (UK) opens the perspective for the domain of unmanned underwater vehicles as an application domain. Their model of autonomous decision making renews the idea of cognitive automation from a different perspective, rounding out the picture.

5.6.1 Artificial Intelligence (AI) Methods Perspective

In order for unmanned air vehicles to fulfil their envisaged enhanced roles in the future integrated battlespace, there is an unprecedented requirement for flexible and autonomous operation, representing a massive leap in capability compared to that of today's systems.

This section critically examines the notions of automation and autonomy on UAV and UCAV platforms, and operator decision support; in the contexts of current and envisaged platforms, roles, responsibilities and missions. The relative merits and maturity of AI techniques and technologies are studied and their applicability to providing elements of autonomous behaviour in UAVs and UCAVs are considered in light of these; illustrating how semi- and fully-autonomous UAVs, and operator decision support, can boost flexibility, survivability and mission effectiveness of UAVs/UCAVs, augmenting and enhancing the capabilities of the future warfighter.

5.6.1.1 Introduction to AI Methods

It has long been recognised, and more recently demonstrated through deployments in the Gulf, Bosnia and Afghanistan, that UAVs have significant potential to enhance a force's ability to project combat power. Their range, persistence, altitude and potential for cost and manpower savings make UAVs an ideal candidate for the dull, dirty and dangerous missions of the future; augmenting, replacing or even surpassing the capabilities of manned aircraft.

Removing the human from the aircraft's cockpit enables more efficient and cost-effective platform designs, albeit with the trade off that the operator is now further separated from the action, with a commensurate possibility for loss of situational awareness. With the operator's role needing to evolve from a piloting to a supervisory capacity, treating many UAVs as one system in order to achieve mission objectives, there are serious implications for cognitive workload [1].

5.6.1.2 UAV/UCAV Autonomy Requirements

Small-scale UAVs such as Micro-UAVs (MAVs), are intended to be deployed by ground forces for short-range surveillance. As such, MAVs are supposed to be man-portable and expendable, hence a fully-autonomous architecture is not required. Much larger UCAV/UAV platforms, although originally intended to be expendable, cannot afford this luxury as development and operating costs continue to spiral upwards. With the human operator removed from the vehicle and reduced to a supervisory role, rather than being in full control, a significant automatic and autonomous element needs to be present for the UCAV to achieve any level of survivability and capability at all above that of a cruise missile.

For a UCAV platform to be successful, it has to rely on achievable technologies, and interactions between component technologies must be carefully managed. Development is concerned with trade-offs between platform properties: individual components are themselves sufficiently advanced already. Constructing the airframe is relatively straightforward; achieving desired levels of survivability and mission effectiveness are the major challenges [2]. How processes are integrated with the platform and infrastructures (e.g., datalinks) is the key.

Currently, UAVs such as X-45A are capable of taking off and landing automatically, and following waypoints to a target. They can relay surveillance data to a controller for target acquisition and designation, and deploy ordnance to destroy a soft target.

The main types of mission that are likely to be undertaken by a UAV or UCAV are listed in Table 5-1 [3]:

Table 5-1: Likely UAV or UCAV Mission Types

SEAD (Suppression of Enemy Air Defences)	Attacking air defences, e.g., SAM sites; plus escort and sweep roles
Strike	Attacking pre-defined targets; plus escort and sweep roles
TCT (Time Critical Targeting)	Attacking targets within a narrow window of opportunity; may involve long loitering times
Maritime specific	AEW, ASW, AsuW tasks; organic to naval vessel or task group
ISTAR (Intelligence, Surveillance, Target Acquisition and Reconnaissance)	Functional requirements including, e.g.: Battlefield Surveillance and Reconnaissance; Target Detection, Location and Identification; BDI/BDA
Communications Relay	Extend LOS communications, e.g., UHF, Link 16

An autonomous UAV/UCAV would have to undertake some of the following actions during each sortie:

- System monitoring.
- Airmanship tasks, e.g., fuel states, aircraft safety.
- Health monitoring/diagnostics, leading to mission re-planning:
  - Work within battlespace infrastructure, i.e., datalinks, communications (manage these); and
  - Fault and damage tolerant control of the aircraft, i.e., it has to fly safely and predictably, work around minor faults, and carry out appropriate procedures in the event of major failures.
- Takeoff:
  - Interoperability with ATC commands, and normal aircraft handling patterns and procedures is vital, even more so for UCAV-N operating from an aircraft carrier (even down to the envisaged level of the air vehicle responding to voice commands from ATC).
- Formation:
  - A need to operate with other manned and unmanned aircraft, in normal airspace, and as part of a package.
- Ingress / Routing to target:
  - Route to pre-defined target;
  - Comply with airspace regulations; and
  - Avoid known and pop-up threats, dynamically altering flight or even mission plans if necessary with SA gained through maintenance of RASP.
- Carry out mission:
  - Compliance with Air Tasking Orders (ATO), SPINS;
  - ROE, Laws of Armed Conflict;
  - Detect, Identify and Acquire Targets- a real-time process for TCT;
  - Weapons release (SEAD, Strike, TCT); and
  - Battle Damage Investigation / Assessment.
- Egress:
  - As ingress; and
  - Added complications for Strike, SEAD, i.e., package re-formation and delousing.
- Landing:
  - Interoperate with airbase landing/recovery patterns and procedures.

5.6.1.3 Applicability of AI Techniques to Mission Requirements

Various AI methods were examined in order to determine their relative merits and particular ‘skills’, when applied to the types of activity necessary for the autonomous mission tasks identified above (see Table 5-2).

Table 5-2: Relative Merits of AI Techniques when Applied to UAV Mission Management Tasks

	Fuzzy Systems	GAs	KBS	NN	CBR	Hybrid systems	MPS[1]
Scheduling/planning	M	M	M	M	Y	Y	Y
Decision support	Y	N	Y	N	M	Y	
Diagnostics	Y	N	Y	M	Y	Y	
Risk analysis	Y	Y	M	Y	Y	Y	
Data analysis	M	Y	M	Y	M	Y	
Monitoring	Y	N	Y	Y	M	Y	
Optimisation	M	M	N	Y	N	Y	
Interpretation	Y	N	Y	M	M	Y	
Classification	Y	M	Y	Y	M	Y	
Control of systems	Y	M	N	Y	N	Y	

[1] Mission Planning Systems.

Legend	
Y	Yes – highly applicable
M	Maybe – slightly applicable
N	No – inapplicable

The following table (Table 5-3) compares the relative merits of these AI techniques to the mission tasks above, giving the tasks fully autonomous UAV or UCAV would have to undertake, and which AI method, or combination of methods would best be suited. These are then linked to various processes (outlined in Section 4 below, and see [4] for more details) forming an example UAV/UCAV architecture (based on the Sharp control and decisional architecture for autonomous vehicles [5]). The result is the identification of which technique or combination of techniques is best suited to each process.

Table 5-3: Recommended AI Techniques for Identified UAV Autonomy Requirements, and Mapping to Processes

Action	Capability	Mission Type	AI task for Autonomous UCAV	AI Technique	Process
Autonomous Navigation	Routing to pre-defined target	All	Route/Mission Planning, Risk Analysis, Optimisation	MPS	Route Planner
	Route re-planning		Mission Planning	MPS	Route Planner
	Mission re-planning		Mission Planning	Hybrid (KBS & MPS)	Mission Manager
Surveillance	RASP from own sensor / datalink → SA	All	Monitoring, Interpretation, Classification, Data Analysis, Optimisation	KBS	Situational Awareness, Mission Manager
	Mission/payload sensor management		Scheduling, Control, Data Analysis	Fuzzy	Mission Manager, Situational Awareness
Threat avoidance	Threat Detection	All	Monitoring, Interpretation, Classification, Risk Analysis	KBS	Mission Manager
	Threat Identification			KBS	Mission Manager
	Threat avoidance through Route/Mission re-planning		Control, Route/Mission Planning	Hybrid (KBS / Fuzzy & MPS)	Mission Manager, Route Planner
Destructive	Target Detection	SEAD, Strike, TCT	Monitoring, Interpretation, Classification, Risk Analysis	KBS	Mission Manager
	Real-time Target Identification		Classification, Data Analysis	Hybrid (KBS / Fuzzy & NN)	Mission Manager
	Target Acquisition		Classification	Hybrid (KBS / Fuzzy & NN)	Mission Manager
	Weapons release		Control	NN	Mission Manager, Flight Management
	Battle Damage Assessment		Interpretation, Classification	Hybrid (KBS / Fuzzy & NN)	Mission Manager
Interoperability	Communications to command centre (maintain datalinks, etc.)	All, esp. Comms Relay, ISTAR, TCT	Data Analysis, Monitoring	Hybrid (NN & Fuzzy)	Mission Manager, Situational Awareness
	Integration into battlespace infrastructure		Data Analysis	KBS	Mission Manager, Situational Awareness
	Fly as part of a package	SEAD, Strike	Optimisation, Control, Risk Analysis	Hybrid (KBS / Fuzzy & NN)	Mission Manager, Situational Awareness, Flight Management
	Interoperate with Airbase / carrier patterns and procedures, ATC	All, esp. Maritime	Control, Planning, Interpretation, Data Analysis	Hybrid (KBS / Fuzzy & NN)	Mission Manager, Flight Management, Situational Awareness
	Interoperate with other manned/unmanned platforms	All	Control	Hybrid (KBS & NN)	Mission Manager, Flight Management, Situational Awareness
System	Fault/damage tolerant control, i.e., flight	All	Control, Risk Analysis, Data Analysis, Diagnostics	Hybrid (NN & Fuzzy)	Flight Management
	Health Monitoring/diagnostics		Monitoring, Classification, Diagnostics	CBR	Health Monitor
	D/NAW capability (aircraft & sensors)	ISTAR	Interpretation (sensor data)	NN	Flight Management, Situational Awareness
	Airmanship, e.g., fuel states	All	Monitoring, Diagnostics	KBS, CBR	Airmanship
	Safety & Emergency procedures		Monitoring, Risk Analysis, Control	KBS	Airmanship

5.6.1.4 Recommended Applications of AI Methods

The analysis in Table 5-3 shows that no single AI technique is appropriate for all areas of autonomous operation. The requirements for all tasks are so diverse that one method cannot be a panacea. A hybrid system would offer the best solution, so that a particular technique that excels in one area can be applied to one subsystem, providing input into other subsystems functioning through one or more different techniques.

By comparing the techniques suited to each task (which were outlined in the section above), and the process that contains each task, it becomes apparent (in Table 5-2) which technique, or combination of techniques is most suited to each process (see Figure 5-30 for a diagrammatic representation).

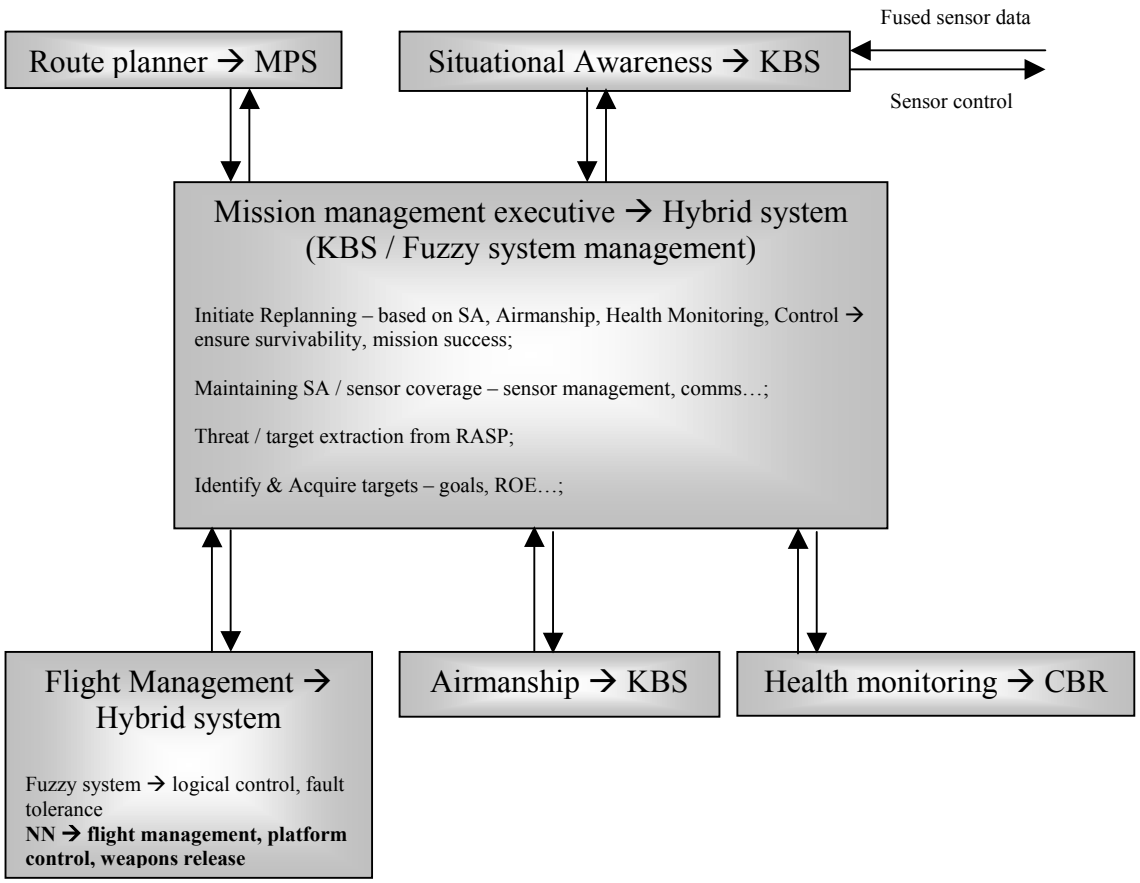


Figure 5-30: Recommended Applications of AI Method.

The **Flight Management** process would be best as a *hybrid* system. A neural network could control the actual platform's flight characteristics, managed by a fuzzy controller receiving instructions from the Mission Manager. This might include neural network for machine vision purposes when the aircraft is operating on the ground (e.g., taxiing around obstacles).

A Case-Based Reasoning system should manage the Health Monitoring process. The present state of the platform would be compared to cases representing nominal operation, and differences used to diagnose failures, and possible remedies, in real time, from a case library.

The Airmanship process would be a Knowledge Based System. This would continually monitor the state of fuel levels, weapons and sensors, and the position of the aircraft with respect to flight levels, safe altitudes, airspace restrictions and so on, advising the mission manager if plans are feasible given current fuel levels, or require re-planning.

Situational Awareness would be provided by another knowledge-based system. This would build a RASP from correlation and fused sensor, datalink, and other information, abstracting tracks, and advising the mission manager on potential threats and targets.

A mission planning system would be best for the Route Planning process. The route planner would take the current mission objective and abstracted RASP from the mission manager and; with knowledge of airspace regulations, the platform's capabilities (from airmanship and health monitoring processes), knowledge of the terrain for masking purposes, and ROE; produce a flight plan.

The Mission Manager is in overall control of the aircraft. This would be use a hybrid technique with inputs from all the other processes feeding a KBS or fuzzy system. A fuzzy system could cope better with uncertain information, however this might be undesirable if information is too vague. The mission manager abstracts information from the SA process into targets and threats, based on mission objectives, ROE and other pertinent information, for the route planner, and passes resulting flight plans to the control process Any changes to the environment, ROE, mission objectives, or RASP may result in re-planning the mission, as could advice from the airmanship or health monitoring processes.

It is important to remember also that a human will still be in the loop somewhere, most likely monitoring the UAV/UCAV as part of a system of systems from a ground station, and possibly taking tactical decisions about the overall mission plan. A soldier on the ground may request surveillance imagery, or even Close Air Support. The overall requirement is for flexibility and network-enabledness as part of the future integrated battlespace.

5.6.1.5 Conclusions on AI Methods

This section has examined the spectrum of available UAV platforms and underpinning technologies in the context of potential scenarios in order to discuss the relative merits of different Ai techniques to improve mission effectiveness.

Of necessity, at this period, most effort has been expended in platform configuration and power plants with bandwidth limitations initially appearing as the first barrier to more widespread application, after the need for greater systems integration.

A trade off then emerges of bandwidth requirements for transmission against onboard processing for greater autonomy, and so reducing reliance on transmitted data.

Autonomy is often seen as a panacea to address the bandwidth problem with AI techniques in the vanguard for achieving this state, although reference to the maturity of different AI techniques quickly refocuses attention on candidate approaches for down selection for any practical demonstrator.

Autonomy and automation are not synonymous, and examples are offered in the report that illustrate the salient differences and the associated impact on elements of any system.

Tabulation of the strengths and weaknesses of different AI techniques indicate the potential application area to best exploit the relevant merits and maturity levels of each technique.

Contrary to widespread beliefs held by non-specialists, this study indicated only limited potential for neural networks, genetic algorithms and fuzzy systems in UAVs, except as a component of more comprehensive hybrid solutions. Fuzzy systems and neural networks were seen as predominantly of use in control and classification applications.

Knowledge-based systems were seen as essential for monitoring, interpretation and data analysis, that is, the less autonomous aspects of missions such as diagnostics and decision support.

The application of specific AI techniques was found to be largely dependent on the level of autonomy envisaged. Higher levels of autonomy suggested more hybrid systems to exploit the complementary capabilities of different AI technologies.

From the systematic consideration of the relative merits of different AI approaches for each UCAV mission requirement, it became possible to identify which AI technique, or combination of techniques are best suited to each element of the system.

Conversely, if different or varying levels of autonomy are considered, appropriate AI techniques can be identified and recommended to satisfy separate elements of the UCAV/UAV system.

5.6.1.6 Recommendations on AI Methods

This study has outlined the relevance and maturity of AI techniques and technologies to address the issue of autonomy in future UAVs and UCAVs. As a result of the process of breaking down missions and roles to address the required functionality it became evident that no single technique was a panacea for complete platform autonomy, however certain methods can be applied to particular areas, which might be self-contained should the goal of complete autonomy be judged unobtainable or undesirable.

The most attainable solution would involve a complex hybrid architecture, utilising several methods to address particular areas, in self-contained elements or as a coherent whole. If a fully or even partially autonomous air vehicle is the goal, then much further development will be required to refine potential architectures in light of implementation issues.

On- and off-board diagnostics and health monitoring could be achieved by a case-based reasoner. A knowledge-based system could address airmanship tasks. Autonomous flight management might be accomplished with a hybrid system comprised of a controlling fuzzy system fed by a neural network. A knowledge-based system is applicable to managing situational awareness, with dynamic route planning an re-planning undertaken by a mission planning system.

Finally, a hybrid system could form the mission management executive, based on a KBS or fuzzy expert system, fed by inputs from the other elements.

There is a vital need to consider more closely the actual architecture of a proposed fully-autonomous UAV in order to gain a better understanding of the particular systems that are likely to be implemented. Then, and only then can a deeper look can be taken at which AI techniques are most applicable to each subsystem; this report identifies the areas that best suit the relative ‘skills’ of certain AI methods.

Any realistic recommendation would be time and funding dependent. Shorter timescales and minimal funding suggest applications limited to diagnostic advisors, whilst longer timescales and more generous funding make possible hybrid AI solutions for applications requiring full autonomy.

AI techniques could also be more confidently employed in areas where less capability is required. This report has identified some of the many issues influencing the path to full autonomy, demonstrating the magnitude and complexity of the obstacles to be overcome. An interim solution could employ intelligent tools to aid the operators of UAVs and UCAVs, both ground and air controlled, to enable them to utilise many more vehicles than at present, whilst working in a supervisory battle management role, as opposed to a pilot role. AI technologies employed in operator decision support, and on- and off-board diagnostics systems could go a long way towards meeting these requirements.

5.6.1.7 References

[1] Draper, M. (2002). Minutes of the first meeting of NATO RTO Task Group HFM-078/TG-017 "Unmanned Military Vehicles – Human Factors Augmenting the Force". Salisbury, UK, 7th – 9th May.

[2] Stolz, S., Capt. UA Air Force UCAV Concept of Operation. Presented at IQPC UCAV Conference, London: 21st – 22nd October.

[3] Nolan II, R.C., Maj; USAF. (1997).The pilotless Air Force? A look at replacing human operators with advanced technology. Air command and Staff College, AU/ACSC/0530/97-03.

[4] Chamberlin, M.R., Thomas, M.J. and Hope, T.M.H. (2003). Assessment of the potential for application of AI techniques to UCAV mission management. QinetiQ Report: QINETIQ/FST/TWP031151. March. UK Restricted.

[5] Vlacic, L., Parent, M. and Harashima, F. (2001). Intelligent vehicle technologies: theory and applications. Butterworth Heinmnann.

[6] Prévôt, T., Gerlach, M., Ruckdeschel, W., Wittig, T. and Onken, R. (1995). Evaluation of intelligent on-board pilot assistance in in-flight field trials. In: 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems. Massachusetts Institute of Technology, Cambridge, MA. June.

[7] Walsdorf, A., Onken, R., Eibl, H., Helmke, H., Suikat, R. and Schulte, A. (1997). The Crew Assistant Military Aircraft (CAMA). In: The Human-Electronic Crew: The Right Stuff? 4th Joint GAF/RAF/ USAF Workshop on Human-Computer Teamwork. Kreuth, GE. September.

5.6.2 Intelligent, Adaptive Help System Design

This contribution describes a comprehensive approach to developing decision support systems for providing intelligent, adaptive aiding to users. The approach is guided by the use of a knowledge design and engineering methodology that combines elements of the CommonKADS and IDEF methods, Explicit Models Design and Perceptual Control Theory. The following sections describe how those individual components should be used in constructing an intelligent help system.

5.6.2.1 The CommonKADS Methodology

"Intelligent" software, such as that required for a truly adaptive help system, is built using knowledge-based systems (KBS), which imitate human reasoning. Such systems are typically composed of formal representations of knowledge about a particular domain and a mechanism that enables the system to "reason" about that knowledge through the application of inferences. This section examines the CommonKADS management and engineering methodology for analysing and designing knowledge-based systems and the role of that approach in developing help systems.

The CommonKADS methodology provides a step-by-step approach to analysing a given problem domain. Questions are posed at each stage through standard "worksheets," which provide a systematic tool for documenting both questions and answers concerning key issues. Individual, numbered guidelines are also offered to assist with the construction of more detailed components of the models.

5.6.2.2 Application of CommonKADS to Help System Design

The process of applying CommonKADS to build a help system involves the specification of the following six models:

- Organisation Model;
- Task Model;
- Agent Model;
- Knowledge Model;
- Communication Model; and
- Design Model.

5.6.2.2.1 Organisation Model

The Organisation Model is assembled during the initial feasibility and assessment phase of a CommonKADS analysis. The primary emphasis of that phase is to examine organisational or business processes that could benefit from the implementation of a knowledge system. Coupled with that is the subsequent feasibility analysis that weighs the costs associated with such an implementation against the projected benefits. Thus, a system must be sufficiently knowledge-intensive to warrant its implementation using CommonKADS. Worksheet questions help to identify the structure of the organisation, relevant processes, people and resources involved and knowledge assets required.

It should be noted that the authors of the CommonKADS approach emphasise the importance of an initial phase in which a systematic analysis of the organisation is conducted. This points to situations where applying the methodology has revealed needs for a knowledge system that are substantially different from those projected at the outset of analysis.

5.6.2.2.2 Task Model

The Task Model in CommonKADS is created primarily as part of the organisational analysis and feasibility assessment and focuses on high-level tasks and goals of agents in the system. Tasks should be examined with a view to identifying those that are sufficiently knowledge-intensive and that would benefit from the implementation of a knowledge-based system.

To create an explicit representation, tasks and sub-tasks should be represented using flow diagrams from the Unified Modelling Language (UML). UML has been adopted by the developers of CommonKADS as the standard technique for schematically depicting activity, state and class diagrams that can represent a wide range of concepts such as data flow, inferences and task structures.

Construction of the Task Model involves the creation of a task hierarchy, in which tasks are decomposed into sub-tasks. Questions posed during Task Model creation relate to constituent sub-tasks, the agents and objects involved, the timing of the task and knowledge required for performing it. That information can then be referred to later during knowledge model construction.

Although CommonKADS specifies that UML be used for representing task hierarchies, there are limitations to UML's expressiveness such that a complement to UML be considered in implementing help systems for certain applications. The limitations relate to the representation of temporal constraints, such as concurrency of tasks and their performance in real-time. In designing help systems for applications in which precise timings of events is critical, it is recommended that the IDEF3 language be used for the graphical representation of tasks and sub-tasks (see the "IDEF3" section, below).

A formal statement of tasks and sub-tasks will help to identify responsibilities that can be carried out by software agents. Such tasks include:

- Monitoring the user's activities;
- Inferring the user's immediate goals and higher-order intentions; and
- Generating system plans to assist the user in the most effective way given current circumstances.

Identification of those tasks should be performed in consultation with subject-matter experts.

It should be noted that the creation of a task hierarchy is fundamental to several of the approaches that comprise the integrated methodology, including Explicit Models Design and Perceptual Control Theory. Task hierarchies are discussed further in sections that follow (see the "Explicit Models Design" and "Perceptual Control Theory" sections, below).

5.6.2.2.3 Agent Model

Like the two models already described, the Agent Model is developed during the initial feasibility phase. It is used to identify the participants in the itemised tasks so that their responsibilities can be incorporated into any resulting knowledge system. That process also assists with identifying expert sources of knowledge that can be useful in supplying information and providing rules for the knowledge base.

Construction of the Agent Model involves examining the tasks in which each agent is involved, the other agents with which it communicates and knowledge that is required to complete its tasks.

5.6.2.2.4 Knowledge Model

The Knowledge Model contains a detailed enumeration of all knowledge required by the system to perform its tasks. Thus, most of the architecture of the knowledge system is designed during the formulation of that model.

The Knowledge Model is subdivided into three categories: "domain," "inference" and "task" knowledge. Domain knowledge contains all of the data used by the application, which, in object-oriented terminology,

would correspond to the class definitions and object instances. In the case of a help system, the domain knowledge should include knowledge relating to the capabilities and functions of the software, including user and system tasks described in the Task Model. Domain knowledge should also include information about the external environment, as contained in the EMD World Model (see the "Explicit Models Design" section below).

The second component of the Knowledge Model is the inference knowledge, which is a collection of methods that act on the domain knowledge. CommonKADS provides a catalogue of inference templates, an approach that has multiple advantages. Creating methods that can be applied generally allows them to be reused readily in other applications. Because they have been applied in other situations, they come pre-tested and therefore contribute to the overall reliability of the knowledge system. Examples of inference knowledge in a help system would include methods for inferring goals from actions or hypothesising a plan based on a user's actions.

The final component of the Knowledge Model is task knowledge, comprising a set of higher-level methods that implement a hierarchy of tasks and sub-tasks. At the lowest level, the sub-tasks make use of methods in the underlying inference knowledge layer, which in turn operate on the domain knowledge. Task knowledge in a help system would include a representation of the full task hierarchy derived in the specification of the EMD Task Model (see the "Explicit Models Design" section below).

As with the inference knowledge, there are templates available for task knowledge and they share the same benefits of reusability and reliability. Templates that are provided include those for planning, scheduling and monitoring, all of which are useful in a help system. Planning and scheduling are tasks required for plan generation and monitoring is essential to plan recognition (see the section on Explicit Models Design below).

Knowledge modelling in any context typically involves the creation of an ontology, and CommonKADS is no exception. Ontologies are formally specified frameworks within which knowledge can be represented. A chief goal in producing an ontology is to identify patterns in the knowledge and exploit those to produce a highly organised and concise specification. One of the main motivations for generating an ontology for a particular domain is to allow its reuse in other applications.

In order to describe fully the Knowledge Model for a help system, it will be necessary first to design the content of the various Explicit Models. The section on "Explicit Models Design" describes how those models are constructed.

5.6.2.2.5 Communication Model

The purpose of the communication model is to describe communication that must occur among agents in the knowledge system. That can include dialogue that is both between the user and system agents, and between individual software agents.

Communication is broken down using a transaction model. For each pair of agents that must interact, a communication plan should be constructed (usually represented using UML) that outlines the flow of information and decisions affecting that flow. That is decomposed further into a detailed itemisation of individual transactions, where each one represents a message sent from one agent to another. Each transaction should be described in terms of the agents involved, the content of the messages and knowledge objects exchanged.

Standard patterns of communication are described in CommonKADS, such as the straightforward "Ask" and the associated "Reply," or slightly more complex exchanges such as, "Require," which can have "Agree" or "Reject" as responses. A library is offered for those and other standard modes of communication.

One type of user-system interaction in a help system involves the presentation of information by the system and possible acknowledgement from the user. (In situations where it would be disruptive to the user to provide explicit feedback, the system should infer through indirect means that the user has received the information.) In addition to that system-initiated communication, users should be able to request help information from the system, which necessitates a second form of dialogue. Interaction may also consist of a clarification dialogue between user and system agents whereby the system seeks information when the user's current intentions are ambiguous, but care must be taken to avoid unnecessary requests for communication with the user. Users may also seek clarification on system goals or activities.

The Communication Model also describes dialogue that occurs among system agents, which is important in the help system to maintain co-ordination among semi-autonomous entities. Agent interaction in the Communication Model is revisited in the "Software Agent Paradigm" section below.

5.6.2.2.6 Design Model

The Design Model examines hardware and software issues related to the construction of the knowledge system. The aim is to take the implementation-independent specifications from the Knowledge and Communication Models and develop a detailed design for constructing the software application, and in the process preserve the structure of those models.

CommonKADS help systems should be designed using the Model-View-Controller (MVC) architecture. In that approach, the Application Model contains the rules, inference functions, and knowledge bases that are responsible for the main functionality of the application. The Views subsystem provides external views of the data in the application model, which can be in the form of a user-interface or can also involve the presentation of information to an external software system. The Controller handles the processing of events, the triggering of tasks and inferences, and the responsibilities of the Communication Model.

The next design step is identifying the target software and hardware platforms. It is recommended that CommonKADS systems be implemented in an object-oriented (O-O) environment.

Some suggested languages for implementing CommonKADS systems are Prolog and Java, but that is not to the exclusion of other possible environments (For examples and source code, see [1] and the CommonKADS web site at www.commonkads.uva.nl).

Once an implementation environment has been selected, the final step in constructing the Design Model is to create a detailed plan for implementation of the Application Model, Views and Controller, as well as the tasks, inferences and domain knowledge within the Knowledge Model. Many details of the plan are dependent on the chosen environment.

CommonKADS also includes guidelines on project management that are designed to accommodate the unique needs associated with knowledge projects.

5.6.2.3 IDEF Standards

The IDEF (ICAM Definition) standards, like the CommonKADS methodology, provide a set of guidelines for analysing processes, activities and information needs within organisations. The IDEF documents are a collection of numbered standards, each of which provides formal guidelines for analysis and design in a particular area. The two standards relevant to help system design are "IDEF3: Process Modelling" and "IDEF5: Ontology Modelling."

5.6.2.3.1 IDEF3: Process Modelling

The CommonKADS approach uses UML for the schematic representation of processes and associated data, agents, tasks and inferences. One of the drawbacks of UML is that it is inflexible in representing temporal relationships and constraints among those elements. Two important temporal concepts are synchronisation and real-time systems. In the case of the former, a distinction is made between synchronous and asynchronous activities, i.e., those that occur at the same time contrasted with those that do not, respectively. That distinction can have important consequences for the system being modelled, where, for example, tasks that can be carried out synchronously can shorten the total time required to complete a procedure and thereby increase the efficiency of the system. When modelling real-time systems, much more attention must be paid to the precise times at which events occur, not simply their relative occurrence, and that can have direct bearing on whether or not the modelled system will perform as intended.

Unlike UML, IDEF3 permits flexible modelling of temporal concepts. For example, symbolic representations exist for depicting whether multiple activities are synchronous or asynchronous and whether all activities must be complete before the next steps in the process can continue.

Modelling of real-time systems is possible using IDEF3's elaboration language, which allows symbolic representations of complex constraints that cannot be depicted using the schematic language alone. The language, based on a subset of the Knowledge Interchange Format, permits formal logical representations of process constraints and allows precise specification of event timings and durations.

Because IDEF3 has full flexibility for temporal modelling, and because it has been in use for a long time, IDEF3 is recommended for use in help system development for applications with precise timing needs. In future, it may be possible to achieve greater flexibility for representing temporal constraints using UML. The upcoming release of the language (UML 2.0) promises to offer more of such capabilities, and the addition of the Object Constraint Language (OCL) to UML also features added support for temporal modelling.

5.6.2.3.2 IDEF5: Ontology Modelling.

Another IDEF standard that is relevant to help system design is IDEF5, which, like CommonKADS, provides specifications for ontology modelling. CommonKADS provides techniques for developing ontologies, including guidance on expert knowledge elicitation, formal descriptions of concepts, attributes and relations, and a formal language for representing ontologies. Although those are powerful components, ontology construction is explored in greater depth in IDEF5, with more guidance offered and more examples provided. The IDEF5 approach has five steps:

- Organising and scoping;
- Data collection;
- Data analysis;

- Initial ontology development; and
- Ontology refinement and validation.

The organising and scoping phase examines the context and purpose of the project, and can make use of the material assembled during the specification of the Organisation, Task and Agent Models of CommonKADS. Data collection involves acquiring raw data by, for example, examining existing systems or eliciting knowledge from experts. Data analysis attempts to refine the raw data into a form more usable in an ontology. Initial ontology development creates a draft of the ontology which is further developed in the refinement and validation phase. The methodology divides each of those five stages into sub-steps and provides detailed guidelines on how to accomplish each.

5.6.2.4 Explicit Models Design

Explicit Models Design (EMD) is a development approach that seeks to make explicit the knowledge required by intelligent software systems. The approach compartmentalises software knowledge into five distinct, interacting models:

- Task Model, containing knowledge (beliefs) about tasks being performed;
- System Model, consisting of the system's knowledge (beliefs) about itself and its abilities;
- User Model, comprised of knowledge (beliefs) relating to the user's abilities, needs and preferences;
- World Model, representing knowledge (beliefs) about the world relevant to the purpose of the software; and
- Dialogue Model, containing knowledge (beliefs) related to communication among human and software agents.

Plan recognition and plan generation are two additional processes that operate within the EMD framework to enhance the software's ability to support the user. Plan recognition seeks to establish the current goals of the user in the context of a larger plan. This process also seeks to recognise goals, plans and actions in terms of the help that might be required by a user to perform a task in a more effective way. Plan generation is used by the system to develop strategies to accomplish its goals, which principally involve providing help to the user. Those techniques, and the individual Explicit Models, are described below in the context of the help system.

5.6.2.4.1 EMD's Contributions to the Help System

Within the help system, EMD offers a means of subdividing the content of the CommonKADS Knowledge Model into components, described in the following sections. Specification of all models must be done in consultation with subject-matter experts.

5.6.2.4.2 Task Model

The Task Model contains knowledge relating to the tasks being performed by the user, represented as a hierarchy of actions, goals and plans. At the lowest levels of the hierarchy are primitive interface actions, such as button clicks and menu selections. EMD recognises that each deliberate interface action carried out by a user is in support of a particular goal and that actions may be expressed in the terminology of such goals. For example, if a user clicks an "OK" button, the system can infer that the user's goal was, "to click the 'OK' button." While the system can easily infer that low-level goal from the simple act of clicking an OK button,

it is typically much more difficult to establish a higher-level purpose unless additional actions are observed. That process, which involves both higher-level goals and context (the particular "OK" button that was clicked), is described in the "Plan Recognition" section below.

Above the primitive actions and their associated low-level goals in the hierarchy are higher-level goals, which can be achieved only by satisfying one or more primitive goals. Sufficiently high-level goals are often associated with what are commonly known as tasks.

A path from a terminal node of the tree up to a higher-level goal constitutes a plan for accomplishing that goal, and there can be many possible plans for satisfying a given high-level goal. Plan recognition enables the system to determine which of those plans a user is pursuing (see "Plan Recognition" below) and plan generation (see the "Plan Generation" section) permits the system to select a course of action from its available plans, or to recommend a series of actions for the user to satisfy an inferred high-level goal.

The tracking of user interface actions and inference of associated goals provides the system with a basis for understanding what a user is trying to accomplish and for helping that user in ways that are both relevant and useful. The system's ability to deduce user goals would be an essential part of any intelligent help system and EMD provides effective methods for designing such a system.

5.6.2.4.3 System Model

The System Model is composed of the system's knowledge about itself, its abilities and the means by which it can assist users. Like the Task Model, the System Model also contains a goal hierarchy, describing the tasks, goals and plans that the system can carry out in support of the user. Those goals are characterised as system support goals.

In the help system, the System Model task hierarchy includes high-level goals, such as, "to assist the user," which would be decomposed into sub-goals, such as those associated with assuming control of functions it had been assigned, monitoring system status and helping the user to complete his or her tasks.

5.6.2.4.4 User Model

The User Model is comprised of knowledge about the user's abilities, needs and preferences. That information is obtained in three ways:

- From information volunteered by the user;
- From results of system requests of the user; and
- From system monitoring of user's activities.

It is worth noting that the system should be able to identify a user so that it can maintain a unique profile for each user. Unless that can be done, the system is reduced to providing information that is often too general, repetitive or useless.

Information volunteered by a user often occurs in the context of specifying options and preferences to the system. It is important that users be able to specify preferences in order to facilitate efficient use of the software, and that is especially useful in applications that offer a large number of features and settings. One method of providing that flexibility is through the establishment of agreements between the user and system using the PACT approach (see "The PACT Approach and Automation Levels" section, below).

There are several ways in which systems can construct user models by explicitly asking questions of a user. For example, if the system has determined which task a user is pursuing, it could enquire whether assistance is needed in carrying out that task. The system also might ask if the user is aware of more efficient plans for accomplishing the task. Finally, if the system cannot determine a user's current plan, it may seek clarification on the user's intentions.

In a truly intelligent system, User Model knowledge is acquired indirectly by monitoring user activities in a Task Model. If the system observes the user carrying out significant and logical portions of a particular plan, it assumes with a fairly high degree of confidence that the user understands that process. The system's confidence increases as the user is observed to repeat the procedure.

If the system determines there is a high probability that a user lacks certain required knowledge, it could signal a need to offer the information. The system must be able to gauge the importance of communicating the information in order to establish a method for doing so. For example, if elements of the total system are at risk of being lost, the user likely would need to be informed immediately. In contrast, advice on carrying out tasks efficiently might not be presented until the user has either completed the current task or finished the session.

5.6.2.4.5 World Model

The World Model contains the software's knowledge about the external world: the objects that exist in the world, their properties and the rules that govern them. Those rules can take on a wide variety of forms, such as physical (e.g., the physical properties of objects in a workspace) and psychological (e.g., rules describing human behaviour in situations of high cognitive workload).

The process of knowledge elicitation required to construct a World Model involves creating a formal representation of information gleaned from subject-matter experts. The resulting compendium of domain-specific knowledge often includes useful information that experts have learned through experience is not contained in existing standard operating procedures. That knowledge can then act as feedback in the review of those procedures.

Knowledge stored in the World Model will form the basis of tutorials and "wizards" guiding the user's pursuit of goals. That guidance may include providing recommendations on creating and manipulating objects, as well as accessing and entering data, activities that frequently require an understanding of how elements within the software interrelate with those in the external world. That knowledge will be structured to support wizards that are adaptable to the range of domains for which help is to be offered.

5.6.2.4.6 Dialogue Model

The Dialogue Model contains knowledge about the manner in which communication takes place among user and system agents. Such communication would involve interaction between the user and system and among other system agents.

Because there are many system agents in a help system, it is essential to specify a common language and protocol for them to communicate. In addition to that, effective user-system and system-system collaboration will require the explicit representations of communication provided by the Dialogue Model.

5.6.2.5 Plan Recognition

The ability to recognise user plans is an important element in EMD and enhances the system's "awareness" of what a user is trying to accomplish so that it can decide how best to offer assistance. It is the infrastructure of

intelligent, adaptive aiding. The COLLAGEN plan recognition approach is recommended for help system implementation.

COLLAGEN uses a "recipe" approach whereby plans that the user may be pursuing are assembled from plan fragments. When an interface action is observed by the system, the fragments are assembled to form alternate sets of plans that might explain why the user performed that action. There may be many possible sets. As further user actions are observed, the number of possible sets that encompass that series of actions diminishes, leading to a more accurate determination of the user's true plan.

Plan recognition should occur in the context of a system of rules to classify user activities according to a set of criteria that identify whether the user is carrying out the current task:

- Correctly;
- Completely;
- Consistently;
- Efficiently; and
- Safely.

Violations of those criteria should signal a possible opportunity for the system to help the user (See "The Five-Part Taxonomy for Plan Recognition" section, below).

5.6.2.5.1 Plan Generation

Plan generation is the process by which the system develops strategies for accomplishing its goals to assist the user. It is based on System Model knowledge of a hierarchy of available support goals and plans, Task Model knowledge of the user's current goals and plans and User Model knowledge of the operator's preferences and abilities.

Plan generation in the help system would seek to construct the most effective plans for offering help to the user, e.g., by displaying a help message immediately or by waiting until a suitable time to present the information with less disruption to the user. System generation of plans also will depend on the selected level of automation.

The processes of plan recognition and plan generation also can be associated with activities in Perceptual Control Theory (PCT), whereby plan recognition is associated with the perceptual input to hierarchies of system control loops and plan generation forms the behavioural output of similar hierarchies. Those parallels bridge the EMD and PCT techniques in the help system (see the "Perceptual Control Theory" section below).

5.6.2.6 Feedback

The concept of feedback is important in EMD for establishing mutual understanding and support between the user and the system, enabling one agent to inform another of its goals, plans and knowledge. Feedback can assume multiple forms, both explicit and subtle.

Explicit feedback can occur in the form of dialogues among agents. For example, the system may ask a user whether he or she is familiar with a particular concept. The user's response constitutes feedback to the system, providing knowledge for the User Model and therefore enabling the system to offer more appropriate

assistance. Similarly, a user might ask the system to explain its last action, particularly if that action was performed on the system's own initiative. The response from the system is feedback that gives the user a better understanding of how the software operates. The communication of explicit feedback among agents is governed by the Dialogue Model, which must be designed to support exchanges among agents involving the provision of feedback.

A less overt form of feedback arises in the form of system support goals and user goals. For example, if a user's goal is to open a window in the software interface, the system will have a corresponding support goal to display that window. The display of that window constitutes feedback to the user that the goal of opening the window in the virtual environment has been achieved. Representations of user goals also are important forms of feedback since they are the primary means by which the system knows and learns about a user. Detecting user goals is detecting feedback in that those goals implicitly inform the system of a user's plans, abilities and preferences.

5.6.2.7 Perceptual Control Theory

Another theoretical approach recommended for use in help system design is Perceptual Control Theory (PCT). IDEF and CommonKADS methodologies provide frameworks for approaching the design and implementation of the system. PCT and Explicit Models Design (EMD) have the potential to influence how the system functions within that implementation framework. This includes how it determines the goals a user is trying to achieve, the plans for achieving those goals and how it can assist the operator most effectively. The techniques complement one another and together provide an opportunity to form a comprehensive approach that combines the strengths of the individual components.

PCT is founded on notions from control theory, in which closed-loop, negative-gain, feedback systems can be used to build powerful models of goal-directed behaviour and to implement complex systems. The ways in which PCT contributes to help system design fall into two basic categories:

- Performing hierarchical goal analyses; and
- Using PCT principles in the algorithms of the system.

5.6.2.7.1 Hierarchical Goal Analysis of Help System Tasks

A method has been proposed for Hierarchical Goal Analysis (HGA) using principles from PCT, and that technique has the potential to produce a robust and complete task and goal decomposition for help system implementation. That approach to systems analysis examines the goal of an agent as a desire to achieve a certain perception.

The PCT-based HGA technique permits two additional analyses to be performed. The first is a stability analysis that identifies possible conflicts among multiple human and machine agents acting on the same system. The second is an analysis of information flow up the hierarchy, which can influence feedback in the system, affecting error-correction at higher levels. Traditional HGA systems analysis techniques consider only the downward flow of information in the hierarchy. In the case of a help system, it is important that information in the form of system and user goals and sub-goals be able to flow freely in both directions. Stability and information flow analyses could contribute to the generation of a robust goal hierarchy for the help system.

A parallel can be drawn with the flow of information within the action, goal and plan hierarchy of Explicit Models Design. When a user is performing actions in the interface, there is a downward flow of information

to which the system responds with system support goals that feed back upward. From the point of view of the system, its goals are met with feedback from the user flowing in the opposite direction. Those properties make the PCT-based HGA approach equally applicable to a parallel analysis within the EMD hierarchies.

5.6.2.7.2 Control Loop Hierarchies

PCT systems can be implemented as a hierarchy of control loops, wherein the output of the higher levels determines the reference signals at the levels below and the perceptions at lower levels feed the inputs at the levels above.

To implement control loops in the help system, the loops need to be assigned a hierarchy of goals, modelled largely at that level and described in the "Hierarchical Goal Analysis of Tasks in the Help System" section above. Those control loops have as input the actions carried out by the user. The actions serve as a basis for system perceptions about the user's need for assistance and that constitutes feedback in the loops.

5.6.2.8 Help System Goals and Sub-Goals

If a high-level goal of the system is, "to have the perception (to believe) that the user is performing the current task sufficiently well, that is, in a way that requires no intervention by the system, a control loop would need to be monitoring activities at the interface (perceptual input) to detect the satisfaction of that goal. Support for such a control loop means that the system must infer and represent a belief that the user is engaged in a particular task based on perceived user activity at the interface. In order to infer such a belief, the system must have knowledge of the structure of goals and sub-goals necessary for operators to perform the task.

The high-level system goal can be further decomposed into sub-goals concerning types of user activity that would suggest to the system whether some form of assistance is necessary. System control loops would identify a need to offer help when it is perceived that the user is not performing a task:

- Correctly;
- Completely;
- Consistently;
- Efficiently; and
- Safely.

That gives rise to five sub-goals, each with its own hierarchy of sub-goals. The system's decision to offer help is based in part on an assessment of user needs according to whether tasks are being carried out in compliance with the five criteria above. For more detail, see the section on "The Five-Part Taxonomy."

In an adaptive interface, system-generated plans to assist users should be incorporated into the behavioural components of control loops. Plan generation mechanisms should examine perceptual error signals and formulate appropriate behavioural responses to correct them. The plan generator should take into account the magnitude of the error signal in determining the optimal behaviour for providing assistance under the circumstances, whether it be automatic or involve querying the operator on how to proceed.

5.6.2.9 Software Agent Paradigm

An autonomous software agent is a programme with the ability to sense its environment and to act on that environment over time to achieve some purpose and to influence what it will sense in the future. Further

distinctions among agents can be made based on their behaviour, for example, communicative agents can interact with other agents or people; adaptive (or learning) agents can alter their behaviour based on past experience; and, mobile agents can move themselves to other machines.

The agent-oriented development paradigm offers several advantages that were not addressed by earlier object-oriented approaches, including:

- Increased modularity;
- Enhanced reusability;
- Improved organisational effectiveness;
- Increased speed;
- Increased reliability; and
- Better distribution.

5.6.2.9.1 Agents in CommonKADS

The CommonKADS (CK) methodology is entirely consistent with the use of software agents. The Agent Model in the methodology allows for systems with multiple human and software components.

(See "The CommonKADS Methodology" section, above)

A Multi-Agent System extension of the CommonKADS methodology (MAS-CommonKADS) has been proposed and is recommended for use in help system development. The methodology was developed to add specific agent-related constructs, including those associated with:

a) Inter-agent communication;

b) The division of tasks among individual agents; and

c) The implications for implementation of multi-agent systems.

The Communication Model in CommonKADS is primarily focussed on interaction between the user and individual system agents, with little attention paid to communication among the system agents themselves. To address that issue, MAS-CommonKADS incorporates a Co-ordination Model, which specifies how messages are exchanged, what communication protocols are used and what abilities each agent has for interacting with others. Because of the many commonalities between the Communication and Co-ordination Models, the latter should be treated as an entity within the former.

The division of labour, allocated tasks, among agents is an important consideration in the MAS-CommonKADS approach. The physical locations of agents and connections among them can influence the assignment of responsibilities to each component. Task allocation also affects the knowledge requirements of each agent.

The multi-agent approach further influences the construction of Design Model specifications. Consideration must be given to network facilities and transfer protocols according to hardware and software constraints.

5.6.2.9.2 Agents in Explicit Models Design

Explicit Models Design (EMD), described above, also supports multi-agent system development. EMD recognises the roles of the User and System as agents and can accommodate both multiple human users and system agents, each represented by its own User or System (Agent) Model.

The EMD Dialogue Model provides a framework for describing communication among multiple human and system software agents. That model allows for various modes of communication, including the following, relevant to the help system:

- A system providing help information and requesting acknowledgement from a user;
- A system prompting a user for clarification feedback about that user's goals; and
- A multi-agent system communicating internally to co-ordinate its overall activity.

As indicated earlier, the System Model in EMD represents the system as a set of co-operating autonomous agents. Provisions are also made for external agents to play a role supporting the goals of both human users and system agents.

5.6.2.10 Hierarchical Goal Analysis (HGA) of Tasks in the Help System

Hierarchical goal analysis (see above, Section 5.4 Perceptual Control Theory, for a discussion of HGA) can be applied to the task goal hierarchy for the help system. In the same way, HGA can also be used in an analysis and design of a goal hierarchy for a network of intelligent agents to assist human users. That analysis offers the same benefits described earlier: a thorough decomposition of the goal-plan-action hierarchy along with stability and information flow analyses.

The highest-level agent network goal could take the general form: "to perceive (believe) that the user is performing the current task sufficiently well," i.e., that the user does not require assistance from any system agent. That goal then can be decomposed into sub-goals regarding the perception of different signs that a user needs assistance, e.g., "to perceive (believe) that the user is performing the current task in the most efficient manner." An error signal would result when inefficiencies in the user's actions are detected, leading system agents to consider intervention.

5.6.2.11 Ecological Interface Design

Ecological Interface Design (EID) is a framework for problem domain analysis and the design of human-machine interfaces in complex work environments. The approach incorporates elements from ecological psychology, particularly the emphasis on the importance of considering the interaction of humans with their environment. While most traditional interface design approaches confine their attention to human characteristics, EID also examines how humans interact with their surroundings, taking into account both physical and cognitive factors, in the context of the complex system under control. To achieve that, EID offers concrete guidelines on interface design, with the aim of producing optimal usability and safety. See [2].

EID should be considered for use in the context of applications that are safety-critical and involve high cognitive workloads.

5.6.2.12 Integrated Methodology for Help System Design

The foregoing sections presented a variety of theoretical approaches to construct a comprehensive, integrated framework for the design and implementation of an intelligent, adaptive, agent-based system for providing

help to software users. The resulting integrated methodology is composed of elements from the following design approaches:

- CommonKADS (CK) – a knowledge management and engineering methodology that guides the systematic analysis and design of intelligent systems;
- IDEF Standards – a complement to the CommonKADS methodology through its more effective support for temporal modelling and ontology construction;
- Explicit Models Design (EMD) – a methodology for building models that identify and compartmentalise the knowledge required by intelligent systems;
- Perceptual Control Theory (PCT) – a feedback control system model for goal-directed behaviour in a system; and
- Software Agent Paradigm – a software design approach that supports enhanced modularity, reusability and efficiency.

The integration of the above techniques into a comprehensive, cross-disciplinary design framework serves the goals of generating a robust, maintainable and reliable help system.

Following is a description of the recommended procedure for designing and implementing a knowledge system within that framework. Steps are designed to be pursued in the sequence presented.

To facilitate the presentation of the procedure, a legend is provided to help distinguish the models that comprise the CommonKADS (CK) knowledge and engineering methodology and those used in Explicit Models Design (EMD).

CommonKADS (CK):	**Explicit Models Design (EMD):**
Organisation Model	Task Model (EMD)
Task Model (CK)	User Model
Agent Model	System Model
Knowledge Model	Dialogue Model
Communication Model	World Model
Co-ordination Model (MAS-CommonKADS)	
Design Model	

5.6.2.13 Help System Design Methodology

- Construct an Organisation Model to describe the organisational structure within which the knowledge system will be used;
- Construct the Task Model (CK), including task hierarchies for all agents identified above (use IDEF3 to represent the hierarchies in applications with precise timing needs);
- Construct the Agent Model identifying all user and system agents and their relationships;
- Generate the Task Model (EMD) by extending the Task Model (CK) to produce task hierarchies for all agents using PCT-based hierarchical goal analysis;

- Develop the User Model according to the need to track user preferences and knowledge;
- Specify the content of the System Model to enable representation and use of system preferences and knowledge;
- Design the World Model to contain required information about the environment necessary for the knowledge system to operate effectively;
- Specify the Dialogue Model, Communication Model and Co-ordination Model to govern the format and content of communication among agents (ensure that the ability exists for agents to provide feedback to one another);
- Use IDEF5 to design an ontology to represent the contents of all Explicit Models;
- Develop the Knowledge Model to encapsulate the ontology and an associated knowledge base containing information from all Explicit Models;
- Within the Knowledge Model, represent the Task Model (EMD) as a hierarchy of PCT loops that use plan recognition and plan generation to form input perceptions and output behaviours; and
- Create the Design Model to produce design specifications for the target knowledge-based system.

5.6.2.13.1 Generalised Principles of Help System Design

Now that the theoretical infrastructure underlying the help system has been described, it is useful to examine some general principles for guiding the construction of an adaptive interface:

- Combine principles from CommonKADS, IDEF Standards, Explicit Models Design, Perceptual Control Theory and agent-oriented development to design and implement the system, as described above;
- Apply the five-part taxonomy within the plan recogniser to classify user interface actions in terms of user help requirements. The taxonomy of requirements includes efficiency, completeness, consistency, correctness and safety, which guide the establishment of a network of rules that enable the system to determine a user's help requirements;
- Use the results of the user needs analysis to develop plans that provide optimal help given the system's on-going knowledge of the user;
- Structure the interface to maximise the effective execution of tasks with respect to the five criteria above;
- Structure the interface to ensure that users receive continuous support in executing plans and achieving goals. A variety of support mechanisms should be implemented, including, among other things, wizards, tutorials and ongoing dialogues between the user and system. Their purpose is to help create a virtual environment where help is integrated, seamless and natural. Not only should those help mechanisms provide guidance and education on software use, but they also should structure the user's pursuit of tasks in a way that allows the system to track user goals and plans with a substantial degree of accuracy. The goal is for the system to take maximum advantage of opportunities to assist users while minimising unnecessary disruptions;
- Implement each object in the software as an intelligent agent responsible for monitoring its status relative to the five help criteria;
- Allow users to specify a "contract" with the system governing the nature of help to be provided; and
- Ensure that help dialogue encourages bidirectional feedback between the user and the system sufficient to mutual understanding.

Those items are described in greater detail in the following sections.

5.6.2.14 The Five-Part Taxonomy for Plan Recognition

In the "Perceptual Control Theory" section, a set of five criteria was introduced for use in classifying whether the user is carrying out the current task:

- Correctly;
- Completely;
- Consistently;
- Efficiently; and
- Safely.

The control loops monitoring user activity should implement rules to determine whether the above criteria are being met. When a failure is identified, the system is signalled (by one or more agents) that the user may require help. A key advantage of that approach is that the rules are not application-specific and can be implemented in any help system that includes, or can construct, a complete hierarchy of tasks and plans for the application. Additionally, most of the help information presented to users as a result of the rules can be derived directly from the task hierarchy (e.g., showing a correct sequence of actions to achieve a specific goal), greatly reducing the need to generate custom help content.

Following is a discussion of the rules that will be associated with each of the five criteria.

5.6.2.14.1 Correctness

Correct execution of plans is associated with carrying out the necessary steps in the required order. There are a few possible scenarios when the system observes a user perform an action that is not the expected next step in the currently inferred plan. For example, it is entirely possible that a user may not know what step should be performed next and has chosen an incorrect action. A user also might repeat a step in a plan, either accidentally or because of a misunderstanding about the correct procedure. Such scenarios provide opportunities for the system to offer clarification on the correct approaches.

Alternatively, a user may have abandoned the current plan, which again offers a possible help opportunity, since the abandonment may be the result of that user not knowing the correct steps required to finish what was begun.

There is also the possibility that a user has deliberately decided not to pursue the original plan, or to follow an alternate plan simultaneously. The knowledge in the User Model relating to the user's expertise, knowledge of particular software functions and past behaviour executing tasks should be organised in a way to facilitate the system's ability to discriminate among such possibilities. Failing that, it may be necessary for the system to ask for clarification on the user's intentions or maintain a level of uncertainty until a clear plan sequence can be identified.

To accommodate such inference capabilities, the plan recogniser must allow for the possibility that multiple plans are being pursued concurrently.

5.6.2.14.2 Completeness

Incomplete execution of plans also can be associated with a variety of scenarios. For example, a user may omit an action while executing a plan, suggesting the system should inform the user of the missed step. A user also might stop carrying out a plan before all the steps are complete, resulting in a need for the system to identify whether abandoning the plan was intentional or inadvertent by examining the User Model and task history, and possibly asking for clarification.

Delays in completing a plan may occur when users are distracted or are pursuing another plan at the same time. In such cases, it may be desirable to present a reminder about the unfinished (suspended) task.

Some actions in certain plans will be identified as optional (either in their ordering or in their presence in a plan) and completeness rules will need to discern whether the omission of an optional step violates the completeness criterion.

It should be noted that in many cases there is overlap between the completeness and correctness criteria, where, for example, if a user skips a step in carrying out a plan, it will be both incorrect and incomplete. However, beginning a plan correctly but not finishing it would lead to a correct, but incomplete plan. In contrast, finishing a plan with an incorrectly executed step might be thought of as a "complete," but "incorrect" plan. Such distinctions justify treating the two criteria separately.

5.6.2.14.3 Consistency

User consistency with task execution can be determined by comparing steps taken to achieve a particular goal with those taken by the user to satisfy that goal in the past. Differences can reveal a number of different things. Inconsistencies in the performance of tasks may indicate confusion on the part of a user as to the correct procedure, presenting an opportunity for the system to provide help. Inconsistencies also may reveal that a user has learned a new plan for achieving a goal, and that new knowledge should be noted in the User Model.

An important point is that an observation by the system that a user is performing a task inconsistently with his or her past behaviour may not indicate that intervention is required, unless there are signs of incorrect or inconsistent plan execution.

A user who varies his methods of carrying out plans may simply be trying to achieve variety in their execution. Knowledge of such tendencies should be stored in the User Model to facilitate future classification of user activity according to the five criteria.

Establishing consistency requires knowledge of past activity and the system will not have that historical data for users who are new to the help system. Because inexperienced users likely will require immediate help, default settings for typical users will serve as a starting point for building a user profile. For intermediate and advanced users, however, the system may conduct a period of observation to become familiar with that user's preferences, needs and methods.

Maintenance of activity profiles should be an ongoing process for both novice and expert users. The system should analyse history information contained in the Task Model for a user to determine preferred plans for accomplishing goals and knowledge of software features. By performing such analyses, historical task data can be archived more concisely and accessed more quickly.

5.6.2.14.4 Efficiency

A simple rule for identifying inefficiency in user actions would compare the current plan with alternate plans for achieving the same goal. The presence of another plan with fewer steps suggests that the user should be informed of the simpler alternative.

There may be situations where alternate plans require the same number of steps, but where one would be considered more efficient. For example, selecting a menu item using a keyboard shortcut is often faster than using the mouse to pull down the menu. Also, some keyboard shortcuts are easier to perform than others, such as pressing the Delete key instead of Ctrl-X to delete an object. A system of rules should take such factors into account and assess the relative ease with which competing plans could be executed.

Another issue is that a particular action may be more or less efficient depending on the other actions that are part of the same plan. For example, if a user has been typing data into a dialogue box, it typically will be most efficient to press the Return key to dismiss the window. However, in circumstances where the user's hand is already on the mouse, it would be easier to click the OK button. Rules in the knowledge system should evaluate the efficiency of steps in the context of the broader plan.

A final rule-based method to increase user efficiency involves identifying plans that the user is pursuing that could be completed by the system without further input from the user. In such situations, the system either could automatically complete the task for the user, or could offer to do so with the user's approval, depending on the selected automation level in the user's contract with the system. In order to offer such capabilities, the help system should be able to distinguish between actions in the goal and plan hierarchy that require human input and those that do not. An example of a task requiring no human input is the ability of most web browser software to enter information into online forms automatically.

5.6.2.14.5 Safety

Safe execution of tasks will not be a critical issue in all applications, but in some domains (e.g., aviation, industrial process control), the safety of humans and equipment can be a deciding factor.

In practically all software applications there is a safety issue surrounding the avoidance of data loss. A simple example of a safety mechanism is the standard prompt to save changes when a file is closed, but more involved methods can be imagined. For example, new users could be presented with additional warnings when deleting complex objects, reverting to default settings or carrying out other tasks where a substantial amount of information could be erased inadvertently. The availability of a "multiple undo" capability helps reduce the risk of irreversible violations of safety criteria.

5.6.2.15 The Five-Part Taxonomy for Plan Generation

The on-going user-needs analysis, described in the preceding section, is responsible for determining whether or not a user requires help with respect to each of the five help criteria. Deciding when and how best to provide that help falls to plan generation techniques.

Generated plans will depend on the nature of a user's need for help. Following is a description of help needs as they fall within the five-part taxonomy.

5.6.2.15.1 Correctness

If a plan is being pursued incorrectly, it is likely that the user should be informed about it promptly. For example, if an incorrect step is performed in a plan, the user will need to know that his goal may not have been satisfied since the plan was executed incorrectly. Thus, the correct procedure for accomplishing that goal should be conveyed to the user promptly and directly, perhaps through the use of an immediate help window.

On the other hand, when there is system ambiguity as to whether a user is carrying out a plan incorrectly, or has simply abandoned it altogether, it will be useful to present that user with a question about her intentions. The help system should allow clarification questions to be presented to users in a non-disruptive fashion, such as through the use of a floating window. Although responding to such questions will be at a user's discretion, doing so will improve the help system's model of that user, and therefore its ability to provide useful help to her.

A user may wish to monitor plans that the system believes she is pursuing and that could inform her that an action is not a correct part of the current plan. That monitoring could occur in a window showing the currently inferred plan in the context of higher-level tasks, as well as what the next action should be. It also could alert the user that the system believes she is pursuing some goal other than her true goal. A mechanism could be provided for the user to correct the system's misunderstanding by specifying the actual plan being pursued, which will provide a learning opportunity for the system. While most users will not find it practical to monitor such a display continuously, it will be very useful in situations where users are unsure of correct procedures. Providing a monitoring mechanism for a user could provide opportunities to understand how such information might be organised and presented more effectively in future. It could serve as an experimental design element with some option for user feedback.

5.6.2.15.2 Completeness

An incomplete plan can be the result of a user omitting a required action during its execution and, as such, it is important to inform the user of that omission. In those cases, a help window should be displayed immediately so that the user may correct the error and achieve the intended goal.

Incompleteness may also result from abandoning a plan and pursuing another in its place. In such cases, it may be important to establish through a clarification dialogue whether the abandonment was the result of a shift in intentions or if it arose because the user lacked required knowledge. The latter case would signal a need to present the missing knowledge. A user might also abandon a plan with which he is unfamiliar in favour of one that he is confident will work. In that case, and under the right circumstances, the system should offer assistance on the initial plan about which he was unsure. Situations where users are pursuing multiple simultaneous plans or have been distracted would not warrant intervention from the system, but provide an opportunity for clarification. Questions seeking clarification should be presented unobtrusively, since it will not always be possible for a user to suspend the current task to respond to the help system. Clarification questions may be presented in, for example, a help system status line, to enable users to monitor system activity. In situations where the user fails to notice the question within a reasonable period of time, the system should alert him to its presence. Care must be taken to make such alerts noticeable but not intrusive and, where possible, context should be considered before displaying the alert. In applications where space is insufficient to present questions in full, a discreet notification may be used, such as the display of an icon. The "Plans for Providing Help" section, below, offers further discussion of methods for presenting clarification questions.

5.6.2.15.3 Consistency

Some inconsistent behaviour on the part of a user may indicate confusion as to how to achieve current goals. Unlike correctness and completeness, the consistency criterion often will not be associated with an overt sign that the user requires assistance and so, typically, will not call for the same immediate presentation of help information. In those cases, it will be more appropriate to use a less overt form of help, such as waiting for a pause in a user's activities before presenting information, or displaying a subtle prompt to tell the user help is available.

5.6.2.15.4 Efficiency

As with consistency, the efficiency criterion usually is not associated with problems in completing the current task and, therefore, an inconspicuous or delayed form of help is also appropriate. Delayed help will presented upon completion of the current plan, at idle time, at the end of the session or under other circumstances that will not disrupt user performance.

5.6.2.15.5 Safety

The safe execution of tasks is associated with a high priority for conveying help to the user. Unsafe acts may or may not involve risk to human safety, but there may be risks of data loss. Help information relating to safety will necessitate immediate notification of the user in a conspicuous manner, such as with an auditory signal accompanying a dialogue box requiring user acknowledgement.

When accidents occur in safety-critical domains, such as aviation, they are typically followed by investigations to identify their attendant causes. An accident occurring in the context of a user interacting with an intelligent system provides an opportunity for investigation results to assist software developers in redesigning systems to prevent future accidents. The software application, Systematic Error and Risk Analysis (SERA) Tool, was developed to help accident investigators identify failures and their pre-conditions using principles of Perceptual Control Theory. Results of investigations using SERA would be useful to designers seeking to build safer software systems.

5.6.2.16 Plans for Providing Help

Help can be offered to users in a variety of forms:

- A "wizard" interface to guide the user through a complex process, such as creating a workspace and configuring its contents (see the next section on "Wizards").
- Tutorials tailored to a user according to what the system believes the user knows. It may be presented according to a prearranged schedule resulting from an agreement (PACT "contract" – see "The PACT Approach and Automation Levels" section, below) between the user and system (e.g., a weekly mini-tutorial on a feature with which the user is not familiar).
- Interactive tutorials, whereby the system steps through a description of a procedure while the user carries it out.
- Tutorials providing guidance on how to solve an application-specific problem that a user is confronting. That would occur, say, in response to a user asking, "How do I finish this [the task the user has begun]?".
- Presentation of a question asking if the user would like assistance or a question asking for clarification of a user's intentions:

Such questions could be displayed in a help system panel within the main application window, and could be accompanied by one or more indicators, such as light bulb icons, to show the availability of messages from the help system. A small panel could show the current question or partial text of that question so a user could decide whether to respond. If the user did not respond within a specified time, the light bulb icon and question would flash two or three times to attract the user's attention. The question would disappear if it continued to be ignored, perhaps after a second round in which the light bulb flashed. Users still should have the option of reviewing that material at their convenience, at the end of a session or in other circumstances deemed appropriate; and

If the system observed a prolonged period without user activity, it might reasonably conclude that the user had suspended activity and was away from the computer. In such a case, the presentation and other system-related activities with respect to help signals to the user would be delayed until work with the software resumed.

- During any review of system proffered help, users would be able to examine the context in which it was determined that help was appropriate, specifically the context of attendant user actions and system-inferred goals and plans from those actions. The display of questions and optional review will be controlled as part of the user's PACT agreement with the system (see "The PACT Approach and Automation Levels" section, below).
- Presentation of web-based help.
- Display of a light bulb or other alert indicating the availability of help information when selected.
- Brief pop-up descriptions of interface elements when the mouse is "over top".
- An offer to complete a task that the user has started.
- Periodic presentation of tips with which the system believes the user is unfamiliar.

5.6.2.16.1 Wizards

To further the goal of creating an interface in which users receive continuous support in executing plans and achieving goals, a variety of wizards should be available to users. The wizard interface approach offers a number of advantages:

- It simplifies tasks for users because the system takes care of detailed plans and contextual supports along the way, making it useful for both novice and expert users;
- Incidental learning occurs when users are guided through procedures for achieving high-level goals;
- By stepping a user through a well-defined procedure, the help system can infer the user's intentions with much greater confidence (especially true for novice users) than if he were proceeding on his own initiative. That substantially increases the relevance and usefulness of help offered by the system about the task; and
- Given the built-in constraints on task execution, there will be fewer opportunities for plans to violate any of the help system criteria.

To help design and implement a wizard, scenarios will be developed to describe how typical knowledge elicitation sessions proceed. That information will be useful in establishing specifications for audio wizards, described next.

5.6.2.16.1.1 Audio Wizards

Consideration should be given to the creation of a wizard interface that uses an audio dialogue between the user and system. Reliable speech tools are now widely available and the incorporation of those techniques into a help system is a logical way to make dialogue more natural between the system and user.

In an audio wizard, questions are posed using software speech generation and spoken answers from users are interpreted using speech recognition. Questions should be phrased to limit the range of possible responses to ensure high recognition accuracy.

5.6.2.17 Control Loop Design

5.6.2.17.1 Centralised System Model Control Loops

Responsibilities for determining user help needs are divided among the System Model and agents associated with individual interface objects. Control loops in the System Model should monitor user actions and apply the general rules identified in the earlier section, "The Five-Part Taxonomy for Plan Recognition." That includes a hierarchy of rules based on those introduced in the "Perceptual Control Theory" section. A high-level description of control loop goals is as follows:

To perceive (believe) that…

- …the user is performing the current task sufficiently well and does not require help;
- …the user is performing the current task efficiently;
- …there are no plans with fewer steps available to achieve the current goal;
- …there are no more efficient plans of the same length to achieve that goal (e.g., shorter time requirements or greater convenience for the user);
- …the user is performing the task correctly;
- …the user has not performed a step that is not in the current plan;
- …the user has not performed steps in the current plan that are out of order;
- …the user has not repeated a step in the current plan unnecessarily;
- …the user is performing the task completely;
- …the user has not omitted one or more steps in the current plan;
- …the user has not suspended the pursuit of a plan;
- …the user is performing the task consistently;
- …the user has not behaved in a manner inconsistent with past task performance;
- …the user is performing the task safely; and
- …the user is performing the task consistent with the integrity of key human, machine and data elements of the total system.

5.6.2.18 Objects as Intelligent Agents

All interface objects in the target software should be implemented as a network of intelligent agents, which has the effect of increasing modularity, organisation and reusability among help system components.

(See the "Software Agent Paradigm" section)

Agent-based objects are to include all standard interface elements, such as windows, buttons and menu items, as well as application-specific objects. Each agent is responsible for monitoring its own status relative to the five help criteria, described earlier. Those include checking the following properties:

- Correctness – e.g., that the value of a variable or the contents of a text box is within an acceptable range;
- Completeness – e.g., that all data have been entered for a particular object;
- Consistency – e.g., that data are consistent with the execution of current tasks;
- Efficiency – e.g., that a frequently used menu item is associated with an easily accessed keyboard shortcut; and
- Safety – e.g., that some agents will have tendencies for self-preservation, particularly where loss of data is a risk.

Agent processes monitoring those criteria should be continuous and System and User Models should be able to query agents as needed.

Agents should monitor the usage of their associated objects by the current user and store important information, such as when they were first accessed, the most recent access and the number of times they have been accessed. That information will be available for compiling User Model knowledge about software features and user familiarity with them.

As noted earlier in the "Centralised System Model Control Loops" section, the System Model should monitor user needs in the context of the general rules of the five-part taxonomy. Specific rules should be associated with interface objects and their agents.

5.6.2.19 The PACT Approach and Automation Levels

Since it is unrealistic to expect that all users will require the same level of automation from an intelligent system at all times, a need exists for users to be able to specify their requirements of the system. In the help system, circumstances of a particular task and operator preferences will dictate what automation level is most appropriate, and those could change over the course of a session.

One method of handling automation levels in the air domain is the Pilot Authorisation and Control of Tasks (PACT) system. That approach is based on the notion of contractual autonomy, in which a user and system establish an agreement, or contract, on the system's responsibilities. Contracts are made using a system of six levels, numbered from 0 (no automation) to 5 (fully automatic). Table 5-4 shows the levels of autonomy in the PACT approach.

Table 5-4: PACT Levels of Autonomy

Levels	Operational Relationship	Computer Autonomy	Pilot Authority
5	Automatic	Full	Interrupt
4	Direct Support	Advised action unless revoked	Revoking action
3	In Support	Advice, and if authorised, action	Acceptance of advice and authorising action
2	Advisory	Advice	Acceptance of advice
1	At Call	Advice only if requested	Full pilot, assisted by computer only when requested
0	Under Command	None	Full

Contracts in a help system should offer users the ability to set the autonomy level and change it at any time during a session, as well as provide the flexibility to customise the provision of specific forms of assistance. That customisation should allow users to request help at specified intervals or to ask that help be provided in a specified form.

In order to aid in deciding the most appropriate level of assistance to offer a user given the selected autonomy level, the system should maintain a set of numerical scores in the User Model to indicate the preferences and needs of the current user. Those act as thresholds in determining when the system should intervene, based on:

- System beliefs that the user possesses relevant knowledge;
- Feedback from the user in response to each help offer from the system, either by using a "Don't show again" button or by ranking the usefulness of the information on a scale of 1 to 10, as in, "How would you rate the value of this help?" The system also should have the ability, at least in some circumstances, to judge the suitability of displaying a message again, even without a user selecting the "Don't show again" option. In most cases, after information has been displayed and acknowledged by the user, the system will infer that knowledge will not need to be presented again, unless subsequent user actions demonstrate that it was not understood or has been forgotten;
- General feedback from the user on preferred types of help, based on online questionnaire responses from the system on favoured techniques. Such a questionnaire could elicit general information on the efficacy of help provided to a user, perhaps at the end of a session. The PACT contract will allow users to enable or disable the feature according to their desire for such a feature;
- Historical system knowledge of techniques that have been most effective at conveying information to that user in the past, revealed by acknowledged help messages and demonstrated task knowledge; and
- State of completion of the current task, so as to avoid unnecessary disruption.

5.6.2.20 References

[1] Schreiber, G., Crubézy, M. and Musen, M. (2000). A Case Study in Using Protégé-2000 as a Tool for CommonKADS. EKAW 2000: 33-48.

[2] Vicente, K.J. and Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. IEEE Transactions on Systems, Man, and Cybernetics, 22, 4, 589-606.

5.6.3 Autonomous Decision Making for an Underwater Unmanned Vehicle

5.6.3.1 Underwater Unmanned Vehicles

Unmanned vehicles have been around for some years in all environments – on the ground (UGV), in the air (UAV), surface water (USV) and underwater (UUV). More strictly, these have been primarily in the category of remotely operated vehicles (ROV), in that there are command and communications links (direct or via satellite) to a remote human operator, who maintains full system control at all times.

In the UAV field in particular, the effort is directed at improved data fusion and representation, in order to allow the human operator to control several UAVs at once, furnishing the vehicles with autonomy over lower level functionality, but nowhere near full autonomy. Underwater ROVs tend to be tethered to a mother ship via an umbilical cord, supplying power and command and communications links [1].

The free swimming UUVs tend to be very small, of limited endurance, with a single specific task, such as inspection of underwater objects [2]. In all these cases the man is being kept firmly in the loop.

However, the envisaged operational environment for military UUVs precludes such an umbilical link and, indeed, for certain mission phases, any sort of communications with the vehicle at all. This will require the UUV to possess the capabilities to perform all the tasks to be performed, from navigation, power monitoring/management, threat identification and avoidance and payload delivery, through to the far less concrete area of high level decision making in the face of high levels of uncertainty.

The power source will also have to be wholly internal to the vehicle. This is currently based on batteries, although it is likely that fuel cells will replace these as the technology improves. Even the latter will only yield a useful energy output of about 400 Wh per kg, with the potential to (possibly) double this figure within the next 5 years. With current UUVs, such as the USS Manta, requiring up to 50 kW for propulsion alone, plus several kW more for sensor operation (e.g., sonar), the size of the problem of supplying sufficient power to allow the performance of any kind of mission becomes apparent.

The addressing of this power management problem, which is usually denoted by HOTEL, What does HOTEL mean? It boils down to answering the question ‘can I do the mission and return to my recovery point on my power reserves?’. This baselining of the projected energy consumption for the whole mission, continually updated during the mission, underpins every other assessment and decision made during the mission.

5.6.3.2 Envisaged Theatre of Operations for Military UUVs

The US Navy envisage the littoral zone to be the most important for UUV operations [3], from mine counter measures prior to a naval assault, through coastal and channel mapping via sonar, to deploying sonars near enemy naval installations to track asset movement and even kill them with torpedoes. This poses certain problems, from vehicle design through to sensor operation. The traditional teardrop shape is the most efficient, from a drag minimisation point of view. However, this is not appropriate for very shallow waters, giving rise to a significant risk of running aground. Here, a thinner, flatter, more Ray-like profile is more appropriate (e.g., Manta), with a higher power consumption as a result. Conventional sonars will have their performance degraded significantly from bottom clutter in the shallows.

There are, however, research efforts directed at biomimetic sensors, emulating the capabilities of the sonar systems of dolphins, which do not suffer significantly from such problems. This will be crucial, as, in all these projected scenarios, the UUV will be totally reliant on a main forward sonar and left and right short range sidescan sonars to prevent it from running aground (or into obstacles), as it follows the coastline or navigates through narrow channels.

5.6.3.3 Approach to Automation and Decision Making

Whenever we automate any process, there are certain risks associated with that automation. Obviously, in some cases these risks are lower than in the equivalent manual process, in others they are higher. The 3 risk factors of primary concern here are those relating to communication (right information at the right time), workload (system overload) and unpredictability (moving outside the zone covered by prior knowledge and experience).

Figure 5.31 provides a schematic representation of the change in adaptiveness vs. levels of human control, automatic control and cognitive cooperation, for these 3 important risk factors of communication, workload and unpredictability. This displays, in particular, the greater susceptibility of human operators to increased workload vs. an automated system and the reverse in the case of increased uncertainty. Any ways in which we can improve the way an automated system can handle uncertainty will be of particular benefit in the context of an autonomous system.

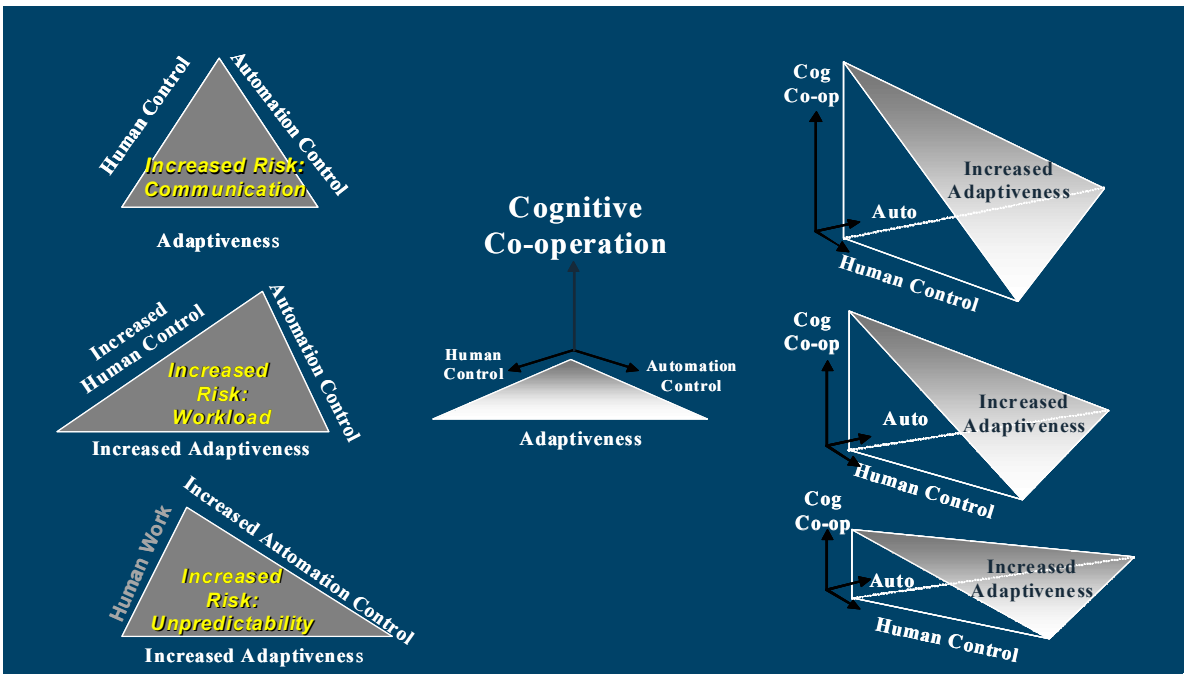


Figure 5-31: Cognitive Co-operation and Human vs. Automation Control.

This is explored further in Figure 5.32, which addresses leveraging autonomy through cognitive automation. This represents the opportunity for pushing the boundary back between what can be achieved in an automated system (traditionally skill and rule based behaviour) and that, knowledge and experience based area that requires human control. This requires the ability to reason effectively in the presence of significant levels of

uncertainty. This can be based on a combination of model based reasoning and a suitable methodology for resolving situations in which no clear decision is forthcoming, due to conflicting objectives, etc. (e.g., neuro-fuzzy or genetic algorithms, e.g., [4]).

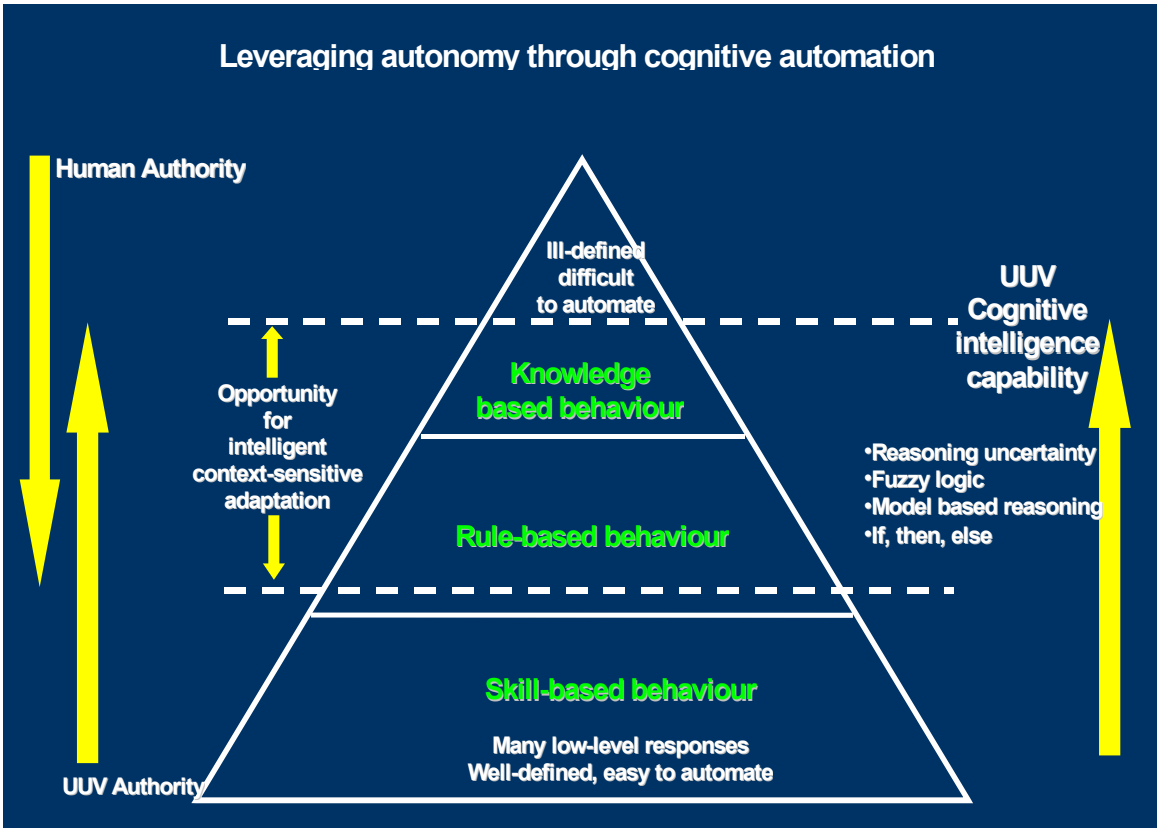


Figure 5-32: Leveraging Autonomy through Cognitive Automation.

The underlying requirement is for a system that can provide for dynamic, context sensitive adaptiveness, so as to engage each task at the appropriate level of autonomy.

It is useful here to simplify the different (cognitive) levels of tasks into 3 groups, namely system health monitoring, mission control and strategic control. Figure 5.33 shows the progression of the dividing line between automated control and human control, from manned underwater vehicles, through the traditionally understood uninhabited vehicle, to an unattended cognitive underwater vehicle, which is an intermediate stage on the way towards our goal of a wholly autonomous (cognitive) underwater vehicle. The aim is to steadily push the dotted line to the top of the triangle, through improved reasoning and uncertainty handling processes. The causal reasoning process can be represented in terms of a decision ladder, consisting of a clear series of knowledge states, linked by processes which determine the flow between these states (a state flow representation).

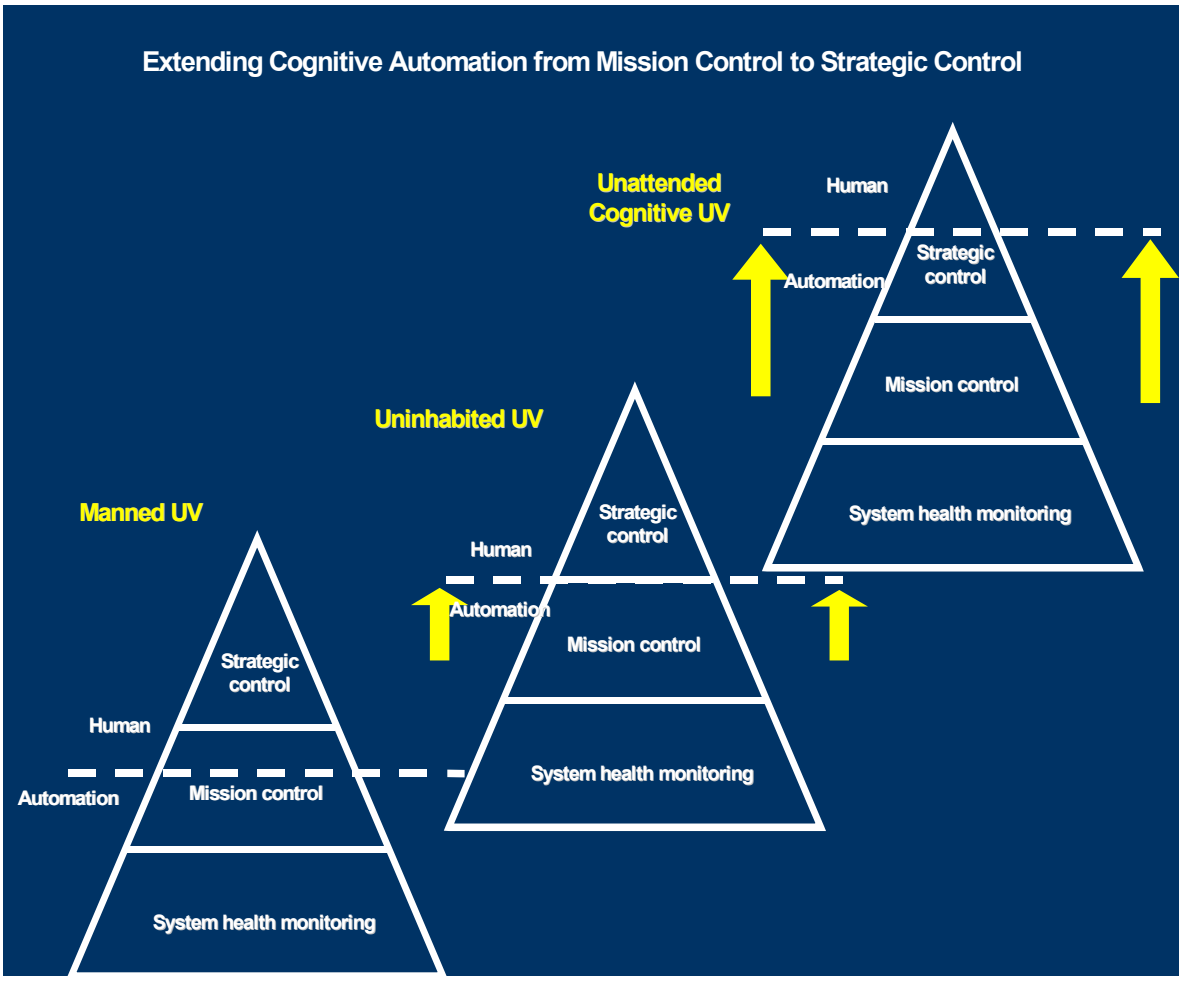


Figure 5-33: Transition from Manned, through Current Unmanned, to Semi-Autonomous Underwater Vehicle.

A general process of adaptiveness is shown in Figure 5.34. Such a state flow representation of a system is, in effect, a finite state machine representation of it and, therefore, by definition tractable. It can be characterised by the traditional measures from information theory (Kolmogorov Complexity [9], Fisher's Information Measure [10], Shannon's Negative Entropy [11], etc.).

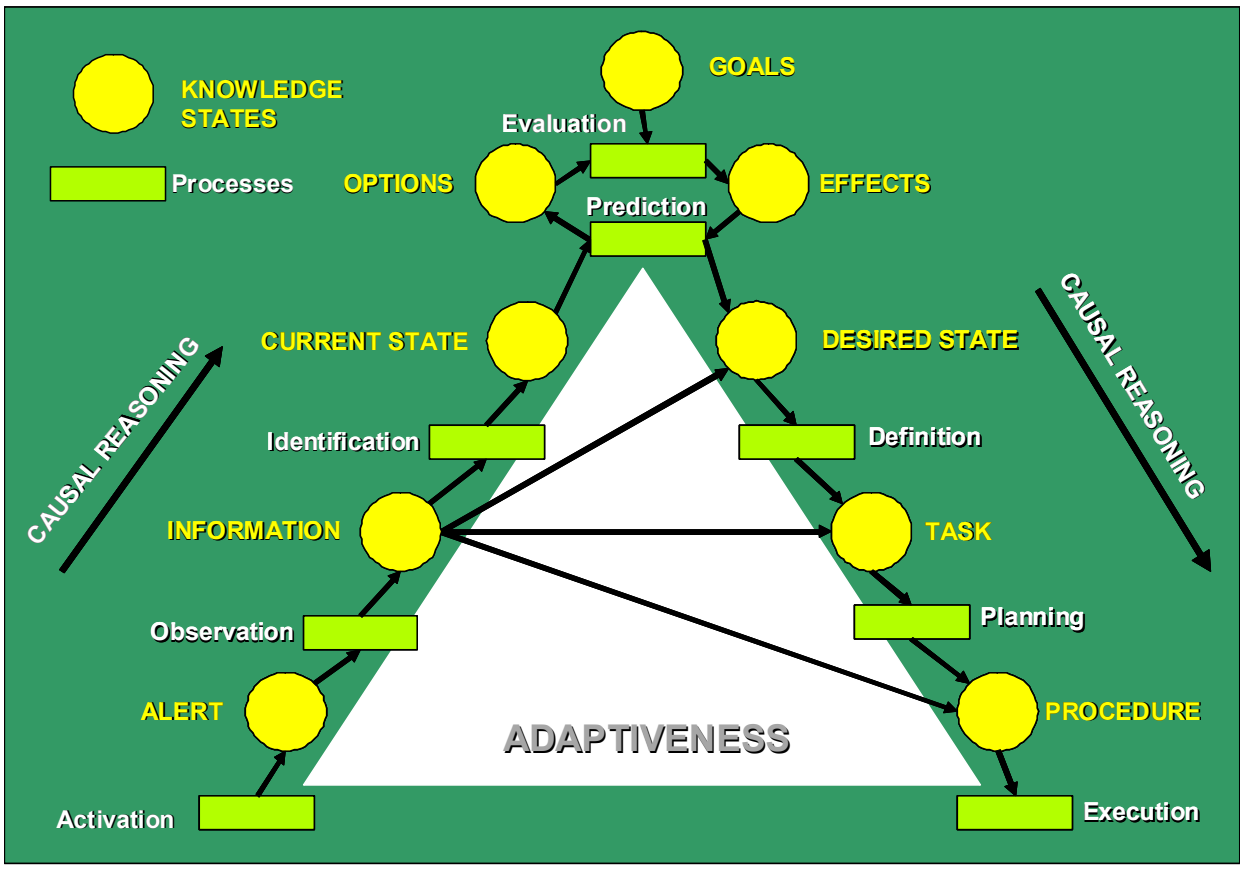
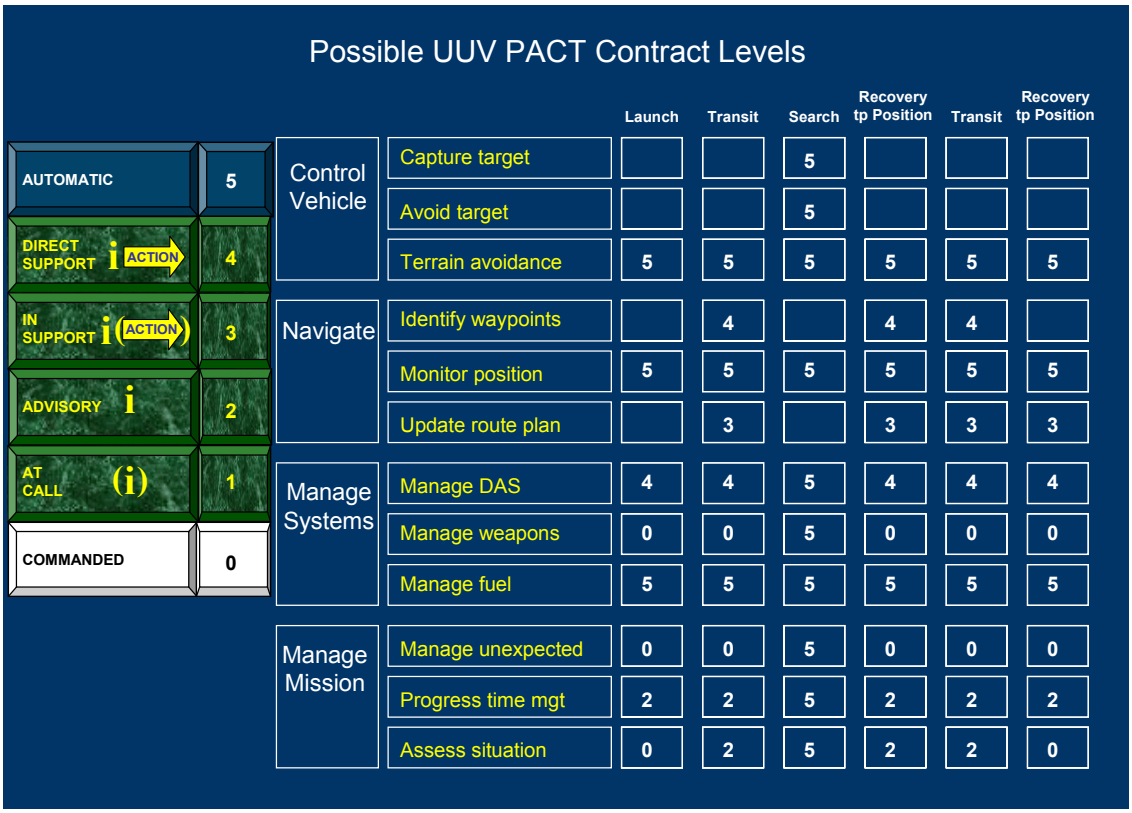


Figure 5-34: Decision Ladder Framework for Task Decomposition.

In particular, the aim is to minimise the Kolmogorov Complexity of the system representation. In effect, minimising this equates to an Occam's Razor approach to decision making and problem resolution. This is defined as the shortest algorithm that fully represents the total information content of a given system. In practice, we will not have the full information content of the system (i.e., mission) and will be operating on a best approximation to that. This arises directly from the, at times considerable, uncertainty that may be present during the course of a mission and any resultant conflicts within the decision process.

PACT (Policy for Authorising and Control of Tasks) provides a convenient system for identifying levels of autonomy in a system (Commanded; Assisted – At Call, Advisory, In Support, Direct Support; Automated). The system is one that has been applied at DERA previously for addressing decision aiding and support in the Cognitive Cockpit Programme [5,8]. The aim in improving the ability of an autonomous vehicle to handle higher level functions and tasks and environments with significant levels of uncertainty associated with them, can be expressed directly in terms of increasing the PACT level of tasks from that currently attainable. As an example, Figure 5.35 provides a possible PACT Contract Level diagram for a set of 'typical' tasks, in 4 groups, over a 6 phase mission for an intermediate type of autonomous underwater vehicle, as in Figure 5.35. Those tasks with a PACT level below 3 do not involve suggested courses of action from the vehicle for the human controller; they require direct initialisation by that controller. The PACT Scheme provides a useful tool in aiding this process, allowing a direct visualisation of the dynamically changing autonomy structures within the system.



		Launch	Transit	Search	Recovery tp Position	Transit	Recovery tp Position
Control Vehicle	Capture target			5			
	Avoid target			5			
	Terrain avoidance	5	5	5	5	5	5
Navigate	Identify waypoints		4		4	4	
	Monitor position	5	5	5	5	5	5
	Update route plan		3		3	3	3
Manage Systems	Manage DAS	4	4	5	4	4	4
	Manage weapons	0	0	5	0	0	0
	Manage fuel	5	5	5	5	5	5
Manage Mission	Manage unexpected	0	0	5	0	0	0
	Progress time mgt	2	2	5	2	2	2
	Assess situation	0	2	5	2	2	0

Figure 5-35: Possible PACT Contract Levels for a UUV Mission.

5.6.3.4 Multi-Agent System Representation

Recent work addressing dynamically adaptive autonomy in multi-agent systems [12,13], puts forward a framework that permits agents to dynamically form, modify or dissolve goal-oriented problem solving groups. In particular, it enables these agents to do so in the presence of high levels of change and, ultimately, uncertainty. It provides them with the key ingredients that permit the transition from adaptability to adaptation – motivation and the ability to change the problem-solving framework of the system itself.

In addition, [12] introduces the concept of the sensible agent, which participate in a two phase process prior to engaging in tasks, namely:

1) A decision-making phase, during which sub-tasks designed to carry out the goal are identified and agreed upon; and

2) A task allocation phase, during which the various agents are assigned actions and tasks, in accordance with the decisions made.

The point is made that agents using any autonomy model must comprehend the concepts of 'self' and 'others'. An example would be the sensor system processor cooperating with the navigation system processor to prevent the vehicle running aground or colliding with an obstacle. To achieve this, each agent must possess its own environmental model, such that it can understand the implications of the information it is receiving for any of the other agents and act accordingly.

A constrained version of this model can be applied to the case of multi-processor platforms, such as UUVs (e.g., Manta), where certain actions must, by definition, be performed by a particular agent (i.e., processor) and where there is a requirement for a single focus for enacting the overall mission plan.

This is most conveniently achieved via the fact that UUVs normally possess a central, or main, processor, which can be permitted to act as arbiter in task assignment and decision-making in general, fulfilling, as such, the role of a local master.

It can also, where operational conditions permit, facilitate the dissemination of relevant information between agents, to augment their common knowledge base (i.e., corporate knowledge). Moving beyond the level of the individual platform, within the context of Network Enabled Capability (NEC), this approach may easily be scaled so as to encompass cooperation among any number of unmanned platforms, in order to achieve a common mission objective.

5.6.3.5 Autonomous Underwater Decision Making

Dstl work has provided a simulation of key aspects of the core UUV functionality in an "unattended cognitive underwater vehicle" (UCUV) autonomy project. This work is intended to provide partial proof of concept demonstration of specific autonomous decision making activities by unmanned vehicles using a navigational exercise. This work is in its early stages of development and there is a long way to go to realise the ultimate aim of a wholly autonomous cognitive underwater vehicle. Figure 5.36 provides an overview of the various inputs and contributing components for this modelling process.

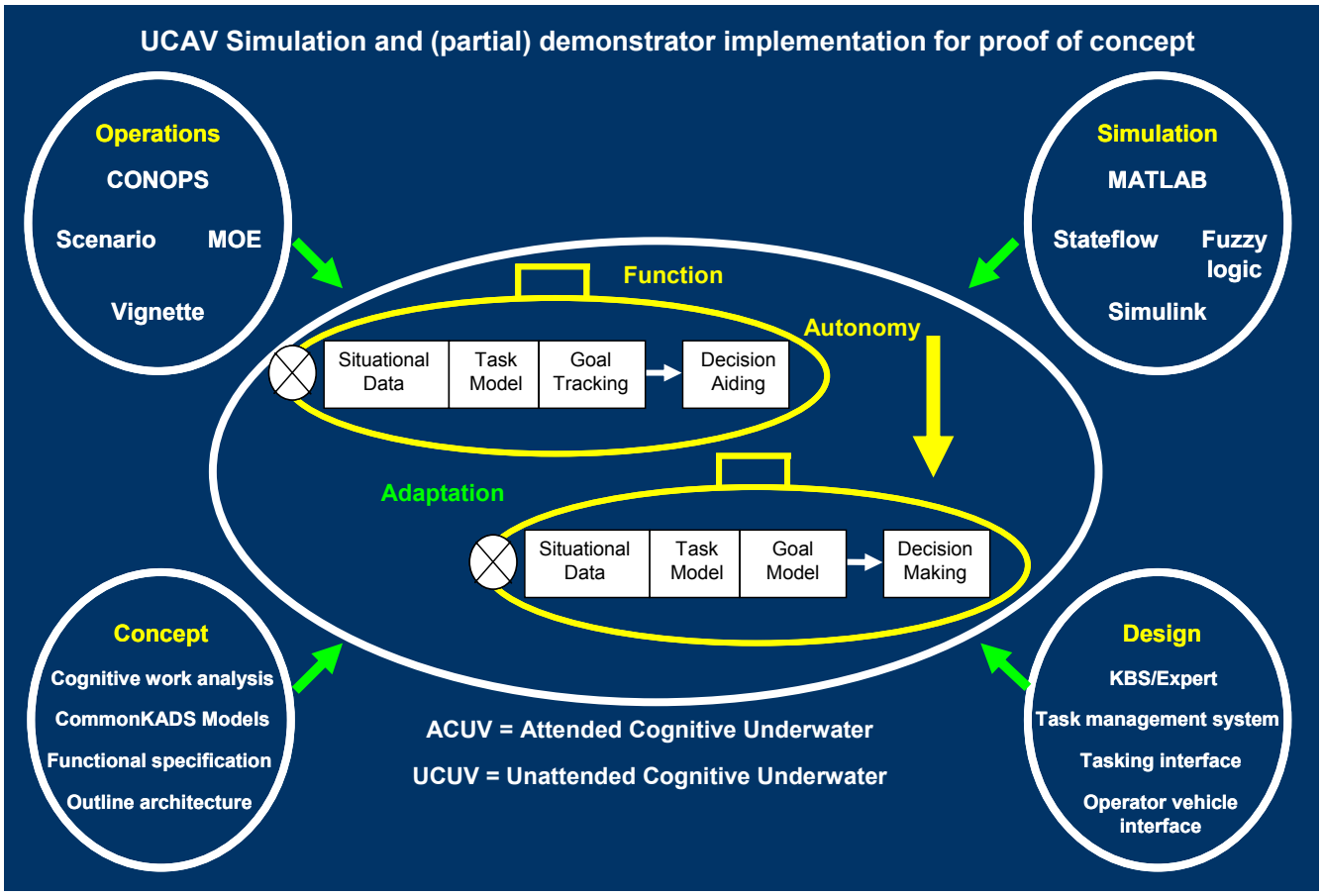


Figure 5-36: Tools and Techniques Drawn upon in the DSTL UCUV Work.

The work on UCUV artificial intelligence engine development has two facets:

1) The development of a knowledge based system for "inside-the-envelope" decision making [14]; and
2) The development of a Bayesian based learning techniques for "outside-the-envelope" decision making [15].

In order to capture a capability to handle 'within the envelope' higher level decision making (i.e., based on direct knowledge), a core knowledge base has been constructed to encompass a subset of the relevant drivers and principles involved [14]. Initial knowledge acquisition (KA) focussed on UUV navigating to a given target (identified by Rapidly Deployable Sonar) whilst avoiding charted objects en route. Using CommonKADS knowledge modelling methodology [16,17], further KA enabled knowledge models to be constructed, from which an intelligent routing application was specified, designed and implemented.

In a fully autonomous system, without recourse to guidance from a human operator, all of the problems that may be encountered must be resolved on-board and an appropriate decision made in all cases. So, where the situation encountered lies 'outside the envelope' for this knowledge base, or where no clear cut decision can be made for some other reason, an alternative decision engine will be needed to apply to the process, to force the issue.

Based upon a survey of the relevant literature, a Bayesian learning based decision engine was decided upon for addressing these "out of envelope" situations [15]. This was adopted because of the inherent advantages of the approach over alternatives such as neural networks, for handling 'out of the envelope' situations, where a traditional expert system type approach is inapplicable. Of particular utility here is its' robustness in the presence of high levels of uncertainty. An incremental variation on this approach, the Bayesian Agent, was also identified, which offers a capacity for both on-line and off-line learning [18].

This survey also identified several priors of potential utility here. In order to evaluate these various options, a simulation exercise was conducted on narrow channel navigation scenarios for a simulated estuary environment.

The work employed the MATLAB modelling environment since this permits a rigorous top-down design methodology to be applied to the construction of the model, for configuration/life-cycle management purposes. This system also permits the pull through into C code of the model components directly from the model itself. Stateflow is a part of the MATLAB suite, which allows the construction of block based models of finite state machines, through the Simulink dynamic simulation system.

A stylised simulated estuarine environment (1000 m x 500 m x 25 m deep), populated with a rough terrain bed and traversed by a simulated narrow navigation channel, cut through the bed to the bottom, was chosen for reconstruction in simulation.

As a result of an extensive literature search, two alternative Bayesian based approaches were selected for comparison (MML and BCPN). What does this mean? Naïve Bayesian learning was employed on the network as a control in the exercise. In addition, an initial investigation was made into the suitability of using an ARTMAP, What does this mean? neural network approach to selection of appropriate prior distributional forms for the system [19]. The problem that was posed to each of the Bayesian systems, was: Given the current direction of travel, provide the system with any changes of direction that are required, so as to maintain a course that is as close to the centre of the channel as possible, at all times.

To, this end a 20 m by 20 m 'window' (i.e., the width of the channel) was used, to simulate a required clearway for safe passage of the craft through the channel. As our system has a 1 m grid size, with each of the mesh points representing the centre of a 1 m by 1 m by 1 m block, this translated itself into a 19 mesh point by 19 mesh point grid, which was required to be maintained clear of solids. For this exercise, no obstacles were placed upon the bed of the channel. However, the trained classifier can be used with equal effectiveness upon the task of avoiding obstacles in the vertical direction.

All three of the Bayesian learning network underpinned approaches investigated can provide, within certain limitations, a means of navigation through narrow, sinuous channels, through the provision of suitable course corrections. The introduction of signal degradation does impact on the performance of all three approaches, but not catastrophically so (graceful degradation). By limiting the required course corrections required to the range of +/- 3 (1 m) grid points per correction, the errors in the corrections supplied all lie with the range of +/- 1 grid point (rounded) after 10 training scenarios, even for relatively noisy data ([0.0,0.4] for 0.0, [0.6,1.0] for 1.0). Due to the much higher computational requirements for the MML approach, coupled with it not providing superior performance to either of the other 2 approaches, it is not considered to be suitable for use for such a task.

The use of an ARTMAP based classifier to select the most appropriate prior form for the nodes in a Bayesian network shows some initial promise. Its use in such a role requires further investigation though.

The BCPNN based approach [20,21] was recommended for pull through into native C code for further testing against a scenario based on real GIS data. This was trained on the training data set, with applied noise levels giving up to 80% reduction in signal separation between land and water, which was that applied to the signals for this test. It was then applied to a compressed section of the Columbia River, providing a tortuous path, with frequent narrowing and shallowing to below the 20 m diameter of the training data channels. This test was successful, with the course followed being within 1 m of the mid line (i.e., 1 grid point).

Overall, the test of the BCPNN approach on real data, based on a section of the Columbia River, with significant 'out of the envelope' behaviour, in terms of narrowing and shallowing, was a success. This shows the robustness of the Bayesian approach itself and also significant promise for the application of the 'abstract and simplify' approach adopted here, in the context of noisy (to 80%), information-poor environments.

The task was extremely challenging since this was a new domain for the application of Bayesian based decision making methods. The methodology is widely applicable, rather than domain dependent.

The overall aim of the work was to provide an initial investigation to provide proof of concept of decision making in support of autonomous UUV operation. This has been accomplished successfully, albeit in a limited way. The project needs to explore and illustrate the use of the developed methodology on a wider subset of the potential problem set.

5.6.3.6 Conclusions

The demonstration of the robust applicability of the principles and approach adopted, to the solution of realistic and relevant problems, certainly indicates its potential worth. This work is innovative in its very nature, particularly within the field of autonomous decision making for unmanned vehicles. It forms an exploration of the possible. However, it is an exploration based firmly on sound mathematical and statistical principles. There is a need to investigate the application of the approach and implemented algorithms to other important noisy and/or information poor decision situations, within the context of projected UUV operational scenarios. Ultimately there is a need to address higher-level decision making tasks in order to provide an integration and a synthesis of the processes, especially within the context of the future network enabled environment.

5.6.3.7 References

[1] Wayamba. (2000). An Experimental Uninhabited Underwater Vehicle. Anderson, B. Paper presented at: Advanced Underwater Technologies for the 21st Century, Tokyo, 23-26 May.

[2] Utyakov, L.L. (2000). Small Sized Underwater ROV. Paper presented at: Advanced Underwater Technologies for the 21st Century, Tokyo, 23-26 May.

[3] US Naval Research Advisory Committee Report (November 2000). Unmanned Vehicles (UV) In: Mine Countermeasures (U).

[4] Wang, P. and McKenzie, E. (1999). A Multi-agent based Evolutionary Artificial Neural Network for General Navigation in Unknown Environments. Edinburgh.

[5] Taylor, R.M., Abdi, S., Dru-Drury, R. and Bonner, M.C. (2001). Cognitive cockpit systems: Information requirements analysis for pilot control of cockpit automation. In: D. Harris (Ed), Engineering Psychology

and Cognitive Ergonomics, Aerospace and Transport Systems. Vol. 5, Chapter 10, pp. 81-88. Aldershot: Ashgate.

[6] Taylor, R.M. (2001). Cognitive Cockpit Systems Engineering: Pilot Authorisation and Control of Tasks. In: R. Onken (Ed), CSAPC'01. Proceedings of the 8th Conferences on Cognitive Sciences Approaches to process Control, Neubiberg, Germany, September 2001. University of the German Armed Forces, Neubiberg, Germany.

[7] Bonner, M., Taylor, R.M. and Miller, C. (2000). Tasking interface manager: Affording pilot control of adaptive automation and aiding. In: P.T. McCabe, M.A. Hanson, and S.A. Robertson (Eds.), Contemporary Ergonomics 2000, pp. 70-74. London: Taylor and Francis.

[8] Taylor, R.M., Brown, L. and Dickson, B. (2002). From Safety Net to Augmented Cognition: Using Flexible Autonomy Levels for On-Line Cognitive Assistance and Automation. In: Proceedings of NATO RTO Human Factors and Medicine Panel Symposium on Spatial Disorientation in Military Vehicles: Causes, Consequences and Cures, La Coruna, Spain, 15-17 April 2002. Paper No 27. NATO Research and Technology Organisation, Neuilly-sur-Seine Cedex (In press).

[9] Li, M. and Vitanyi, P. (1997). An Introduction to Kolmogorov Complexity and Its Applications, 2nd Edition. Springer, Verlag.

[10] Fisher, R.A. (1925). Statistical Methods for Research Workers. Edinburgh.

[11] Shannon, C.E and Weaver, W. (1949). The Mathematical Theory of Communication. University of Illinois Press.

[12] Barber, K.S. and Martin, C.E. (1999). Agent Autonomy: Specification, Measurement, and Dynamic Adjustment. Proceedings of the Autonomy Control Software Workshop at Autonomous Agents, 8-15 May, Seattle, WA.

[13] Barber, K.S., Goel, A. and Martin, C.E. (1999). The Motivation for Dynamic Adaptive Autonomy in Agent-Based Systems. University of Texas Report no. TR99-UTLIPS-AGENTS-02. Austin, Texas.

[14] Thomas, M. (2003). Bayesian-Based Unmanned Underwater Vehicle Knowledge System: Scenario Definition, Application, Design and Implementation. QINETIQ/FST/CR035464. October. QinetiQ, Farnborough, Hants.

[15] Water, M. and Taylor, R.M. (2004). Bayesian-Based Autonomous Decision Making in Unmanned Underwater Vehicles: Scenario Definition, Method Development, Implementation and Evaluation. DSTL/CR11773 V1.0. 07 October. Defence Science and Technology Laboratory, Farnborough, Hants.

[16] Tansley, D.S.W. and Hayball, C.C. (1993). Knowledge Based Systems Analysis and Design. Prentice Hall.

[17] Shadbolt, N.R. et al. (1999). Knowledge Engineering and Management. MIT Press.

[18] Sanguesa, R. and Cortes, U. (1998). The Bayesian Agent: an incremental approach for learning agents working under uncertainty. Dept. of Software, Tech. Univ. Catalonia.

[19] Anagnostopoulos, G.C. and Georgiopoulos, M. (2001). Ellipsoid ART and ARTMAP for unsupervised and supervised incremental Learning. University of Central Florida.

[20] Johansson, C., Raicevic, P. and Lasner, A. (2002). Reinforcement Learning Based on a Bayesian Confidence Propagating Neural Network. Dept. of Numerical Analysis & Computing Science, Royal Institute of Technology, Sweden.

[21] Wallace, C.S. and Freeman, P.R. (1987). Estimation and inference by compact coding. Journal of the Royal Statistical Society, 49(3):240-265.