Defence and Civil Institute of Environmental Medicine, Canada

Principles for Intelligent Human-Computer Interaction

(Paris, Sept 7, 1992)

Martin Taylor
(Then at: Defence and Civil Institute of Environmental Medicine
Box 2000, North York, Ontario, Canada M3M 3B9
mmt@ben.dciem.dnd.ca)
Currently(2012)
Martin Taylor Consulting
mmt@mmtaylor.net

This talk is about the principles that underlie Human-Computer Interaction, especially the interactions we hope to achieve with the "intelligent" computers of the future. It is built around a theory called Layered Protocol (LP) Theory, which can be seen as an application to communication of a general approach to psychology called Perceptual Control Theory (PCT).

Human-computer interaction is seen as an instance of communication between "intelligent" entities, where "intelligent" means not so much cleverness as a certain degree of independence of sensing and of action.

Defence and Civil Institute of Environmental Medicine, Canada

Part 1

General Introduction

The talk is in twelve parts, divided into two sections. The first section (Parts 1–4) consists of introductions to the various concepts of Perceptual Control Theory and Layered Protocol Theory, while the second (Parts 5–12) elaborates some of the concepts and illustrates areas in which they can be employed.

Part 1 of the talk introduces basic ideas about communication and action in the real world and lays out the overall plan of the talk.

Acknowledgments: This talk owes much of its structure and many of its concepts, to David A. Waugh (Andyne Computing Ltd., Kingston, Ontario, Canada). Perceptual Control Theory, a very powerful approach to psychology, is due to William T. Powers, and is well described in his book *Behaviour: the control of perception* Aldine De Gruyter, 1973 (2nd Ed. Benchmark Publications, 2005).

1-2

Defence and Civil Institute of Environmental Medicine, Canada

Basic Principles

Psychology

All behaviour is the control of perception

Layered Protocols

All communication is the control of belief

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Perceptual Control Theory (PCT) and Layered Protocol Theory (LPT) are based on parallel principles. For all questions of psychology, the basic principle is that behaviour exists only as the control of perceptions. A complex of perceptions can be labelled as "belief." When the behaviour is for the purposes of communication, the originator of the communication is trying to control his or her own beliefs about the beliefs held by the partner.

1-3

Defence and Civil Institute of Environmental Medicine, Canada

Reasons for communicating

Communication is always to alter the partner's information state,

which may result in one or more of

1. (INFORM) A change in one's own belief about the partner's beliefs, which can be used to assist later communications.

2. (COMMAND) An action by the partner that affects the world outside the dialogue

3. (REQUEST) The provision of information by the partner.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

A communication has three possible main purposes, all of which depend on the beliefs held by the recipient that are to be changed by its receipt. If the originator's intent is to believe that the recipient knows something or to find out something the recipient knows, the communication is to "inform" or to "request." If the originator wants to perceive the recipient doing something, the recipient's beliefs must be changed in such a way as to get him or her to perform the desired action ("command").

1-4

Defence and Civil Institute of Environmental Medicine, Canada

Communication is:

to change the partner's view of the world	(to inform)
to get the partner to do something	(to command)
to acquire information from the partner	(to request)

The partner is a dynamic entity, hence:

The partner's dynamic state must be modelled if a communication is to have its desired effect.

The choice of messages to effect a communication depends on the models held of the partner, including the partner's view of one's self, of the task, of the dialogue, of the world, and so forth.

and therefore:

Meaning does not reside in the physically detectable messages that can be seen or heard by a third party.

Close friends, prisoners, members of "secret" societies all use this fact to pass undetected or uninterpretable messages.

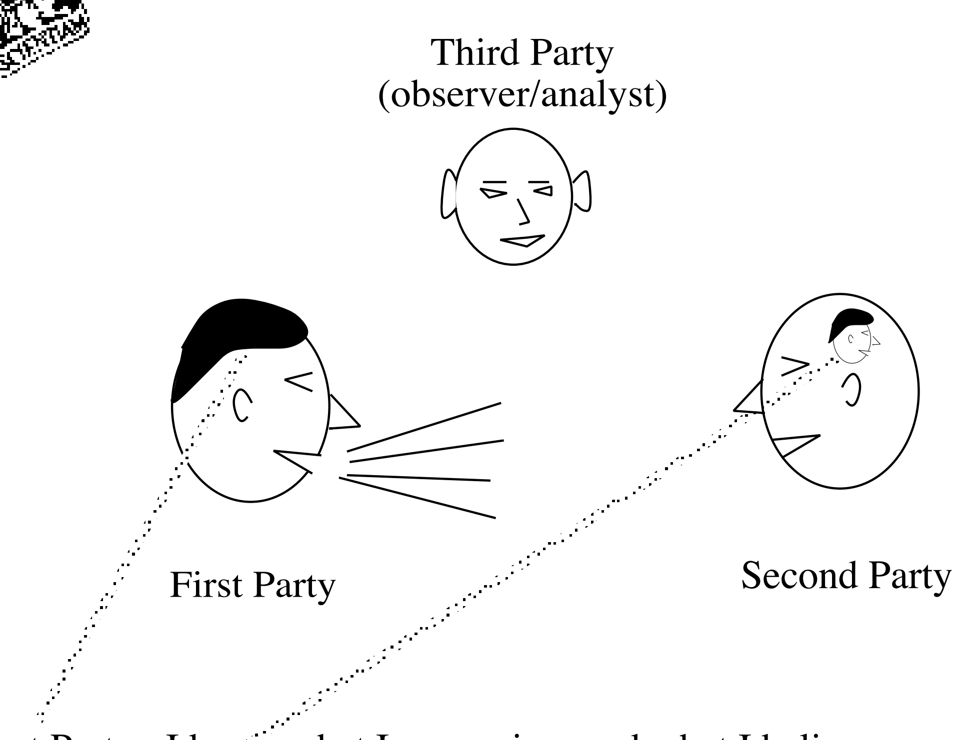
M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The content of the message (its "meaning") is contained in the belief changes of the recipient, as perceived by either partner, and is thus inaccessible in principle to any outside observer. Third parties may, however, infer the belief structures of the participants, and thereby create their own interpretation of the message meaning. This "third-party" meaning may be quite different from the meaning of the message perceived by the participants.

1-6

Defence and Civil Institute of Environmental Medicine, Canada

The "Third Party" Problem



First Party: I know what I am saying, and what I believe.

Second Party: I believe X about what he is saying to me and about what he believes.

Third Party: I can guess what they believe because I can hear what they are saying, and from my guesses I can infer something about what they are communicating.

A "first party view" refers to "my" models; a second party view refers to "my" models of "your" models. Neither partner has a view of the analyst/observer, so communications are not tuned to what the third party might know or believe.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

We use the terms "First Party," "Second Party," and "Third Party" to refer respectively to the originator's view of the world (including the partner), the view of the world taken by the partner, and the view taken by a non-participant in the dialogue. The third party is not considered in the construction of messages, so that the effects of a message on the third party are unpredictably related to their intended effect on the recipient. The third party can determine neither the intended nor the received meaning of the communication.

1-5

Defence and Civil Institute of Environmental Medicine, Canada

Three Independences of "intelligent" communication

1. **Independence of design**
   The coder of one partner cannot be sure how the other's decoder will interpret any message
2. **Independence of input.**
   Neither partner can be sure of what the other knows.
3. **Independence of action.**
   Neither partner can be sure of all that the other is doing, and hence of the other's state.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

If the communicating partners had complete knowledge of each other's state, there would be no communication problem. The originator of a message would know exactly what information was needed in order to change the recipient's belief state to its desired condition, and could construct a message that was encoded in a way that the recipient would be sure to interpret correctly. Without this knowledge, the originator cannot know precisely what content to include or to omit, and cannot be sure how to transmit that content with assurance that it will be interpreted in the intended way. The communication must become **intelligent**, rather than being just a process of encoding and decoding according to agreed rules.

1-7

Defence and Civil Institute of Environmental Medicine, Canada

Partner Models

in fiction

Professor Moriarty, "The Napoleon of Crime," visits Sherlock Holmes, the two having never previously met...

Moriarty: You evidently don't know me.
Holmes: On the contrary, I think it is fairly evident that I do. Pray take a chair. I can spare you five minutes if you have anything to say.
M: All that I have to say has already crossed your mind.
H: Then possibly my answer has crossed yours.
M: You stand fast?
H: Absolutely!

(A.Conan Doyle: *The Final Problem*)

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

1-8

Defence and Civil Institute of Environmental Medicine, Canada

Partner Models

in practice

In a simple (Macintosh) interface modification, the computer models the user's behaviour.

When the user asks to open a file, the Macintosh normally displays a list of files available in an "active" folder. If the user wants another, the folder hierarchy must be navigated. The modification allows the user to select a different folder from among the ones most recently accessed. The computer models the user as being likely to work with files from a small set of localities. What the user is likely to say has "already crossed the computer's mind."

(Actually, it is the interface designer who models).

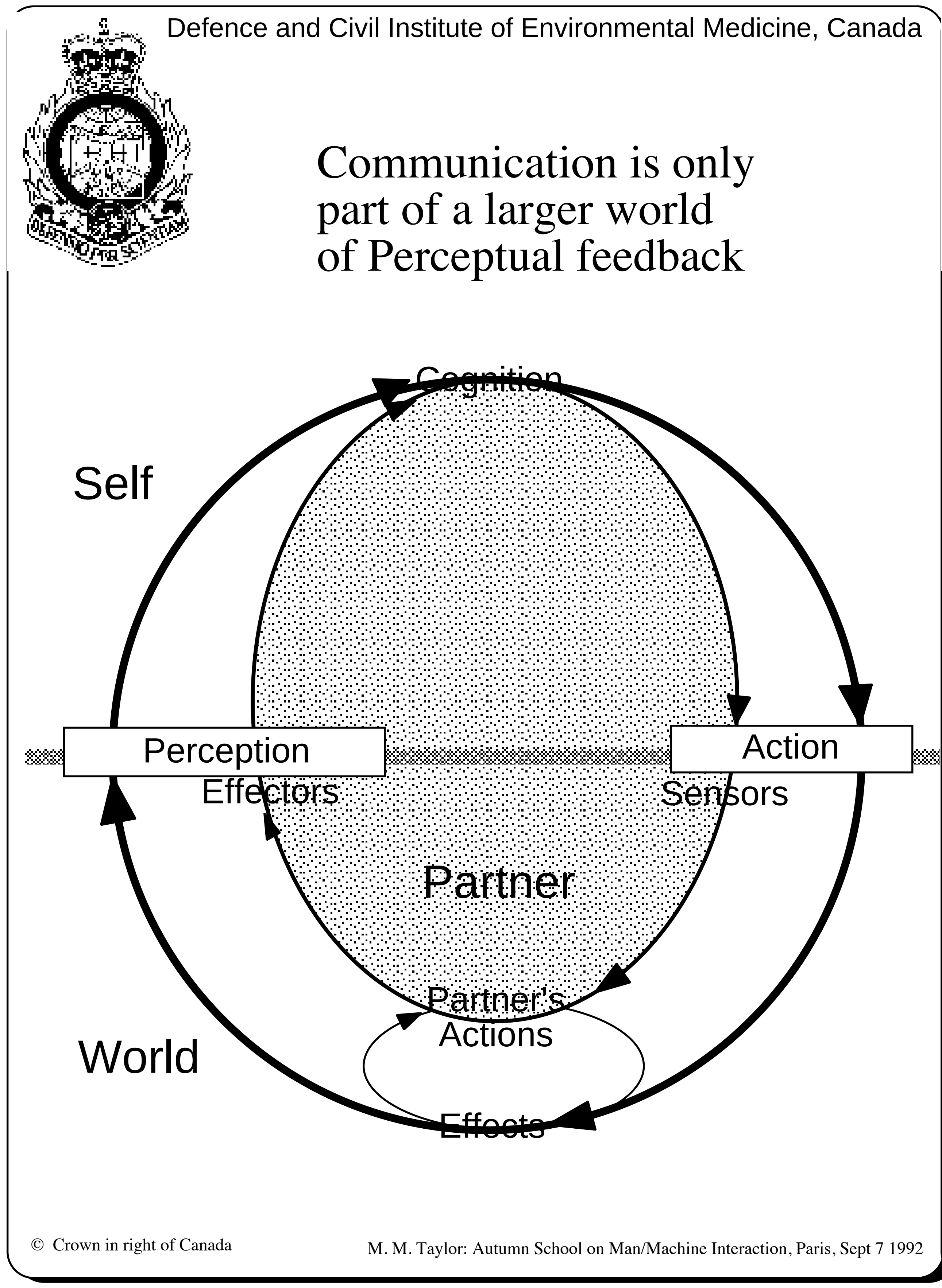
© Crown in right of Canada

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

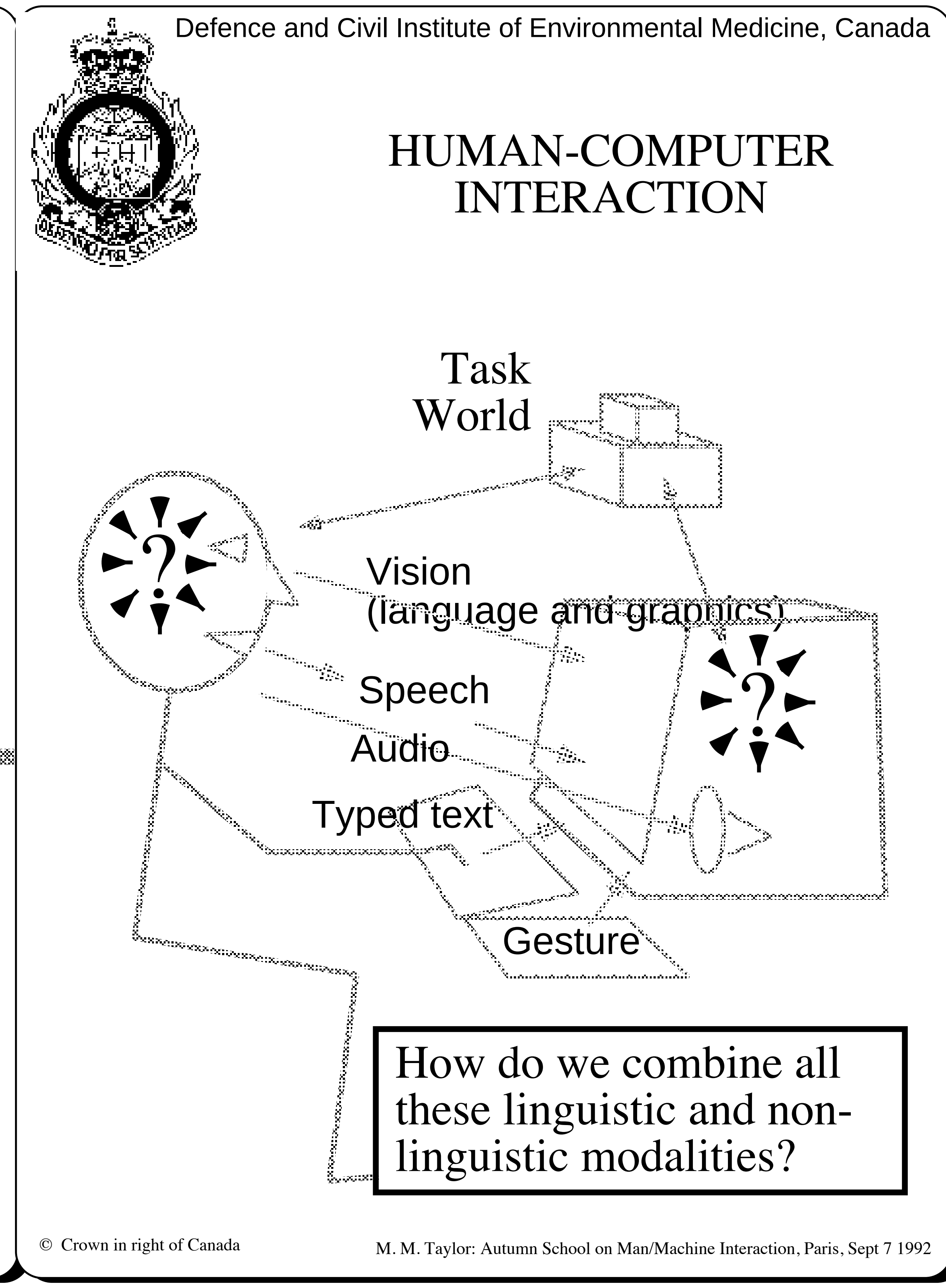
Effective partner models enable the participants to minimize the amount of low-level transfer of information. A few words may be enough to convey a message that would take a long argument if the models were less complete or less accurate. Sherlock Holmes had long studied Professor Moriarty and his methods, and vice-versa. Each was expert in analyzing the behaviour of strangers, using minimal overt cues, and needed almost no words to complete a negotiation that would be totally obscure to a naive observer. Such models must change as circumstances require that they be updated. The ongoing dialogue is one source of information that affects the updating procedure. Moriarty initially was not sure whether Holmes could be persuaded, but now can add extra firmness to his model of Holmes' behaviour.

At the interface, effective models permit the user to execute desired actions easily and with few overt acts. One frequently effective element of a model is that users often want to use files that are contained in folders recently accessed. The original Macintosh interface took no account of this fact, except to provide as a default folder the one last accessed. By adding a list of the ten most recently accessed folders, the modification (a system extension) saves many users a great number of explicit specification acts, without adding to the difficulty of specifying a folder that was not recently accessed.

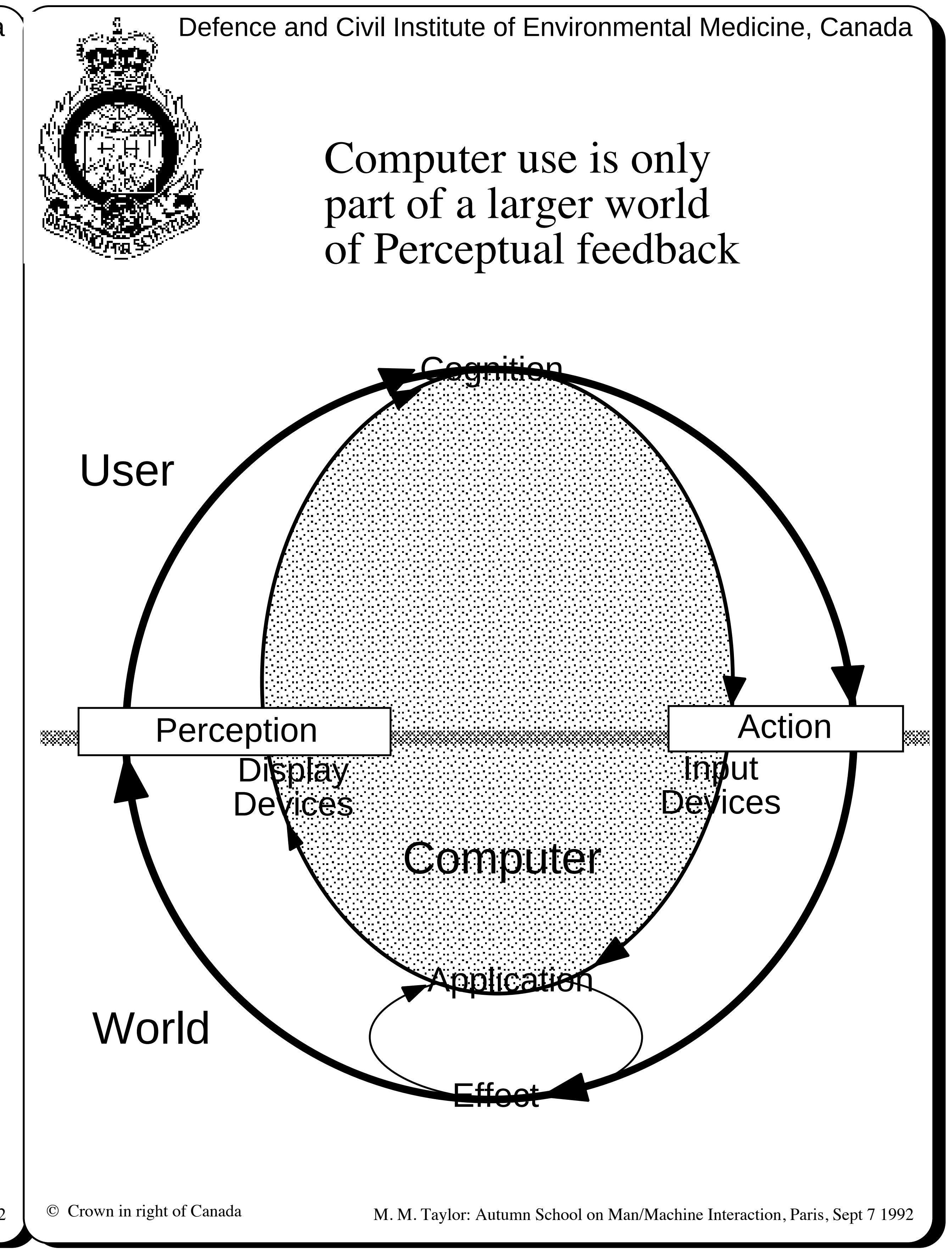
1-9



1-10



1-11



All actions are situated in a world that is perceived through various kinds of sensing mechanism, and the actions are performed only so as to make perceptions approximate some desired states. Some of the desired effects on the world are most easily performed with the aid of a partner, which is why communication exists. There are two kinds of perceptions: one being of the partner's actions and the other being of effects in the rest of the world, some of them caused by the partner, some caused directly by the original actor.

When the partner is a computer, its sensing mechanisms include many possibilities, both linguistic and non-linguistic, but the basic principle is the same. The user is controlling perceptions that are affected by his or her actions either directly or through the mediation of the computer, and some of those perceptions are of the actions (outputs) of the computer itself.

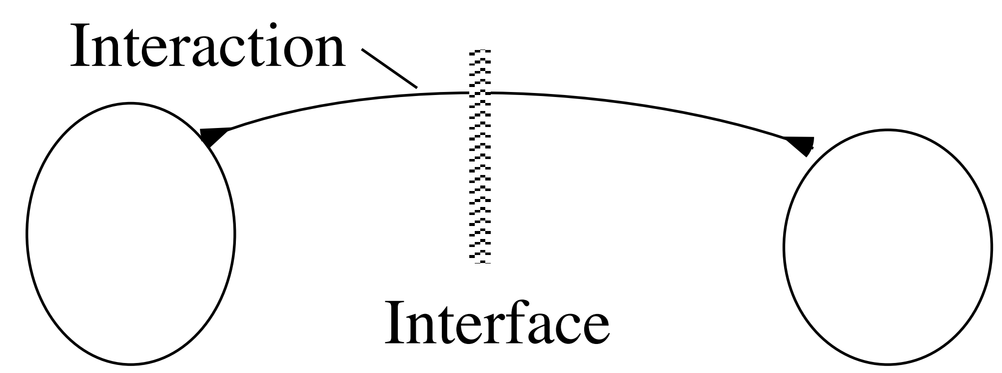
1-12

Defence and Civil Institute of Environmental Medicine, Canada

Interface vs Interaction

Interaction is an ongoing process of information interchange between partners.

An Interface is a structure at the boundary between two partners, through which interaction can occur.



M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

There is a lot of confusion between the words "Interaction" and "Interface." In our work, we think of an interaction as being something that goes on over time between two communicating partners, whereas an interface is a kind of notional surface between the two partners. An interface is a structure, whereas an interaction is a process that uses the interface.

A designer can design an interface without designing the interactions that use it, but the interface constrains the kinds of interaction that can occur. If the designer, instead, designs the interactions, those designs place constraints on the interface without specifying it in detail. Some of these constraints may not be realizable in practice, which enforces an interplay between the design of interactions and of interfaces.

Normally, the design of an interface is done with at least implicit understanding of the interactions that it will support, and the design of interactions assumes some kind of interface that will support them. This interplay is so tight that the two concepts (interface and interaction) are often taken to be the same.

1-13

Defence and Civil Institute of Environmental Medicine, Canada

To do is to perceive; To say is to believe.

Perceptual Control Theory

and

Layered Protocol Theory

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Our descriptions of behaviour and of communication are based on two closely related theories. Perceptual Control Theory (PCT) has been developed by W. T. Powers over the last 40 years, and is well introduced in the book *Behaviour—the control of Perception* (Aldyne, 1973). Layered Protocol Theory (LP) has a separate history, but can now be seen as a specialized instance of PCT that focuses on the interactions among two partners.

LP considers communication as happening in a series of layers. Each layer supports the passage of one-way messages between two partners, along with the associated feedback that enables the correct reception of those messages. These layers are not at all like the layers often identified in other layered approaches to Human-Computer Interaction.

1-14

Defence and Civil Institute of Environmental Medicine, Canada

Perceptual Control Theory

Perceptual Control Theory (PCT) is an integrative theory that covers all aspects of psychology.

PCT addresses WHY we do what we do, based on an evolutionary need to survive. It is based on goals and results, not on sense data and actions.

PCT addresses HOW we do what we do, based on the need for an organism to determine whether its behaviour has survival value. It is based on feedback control systems.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

1-15

Defence and Civil Institute of Environmental Medicine, Canada

Layered Protocol Theory

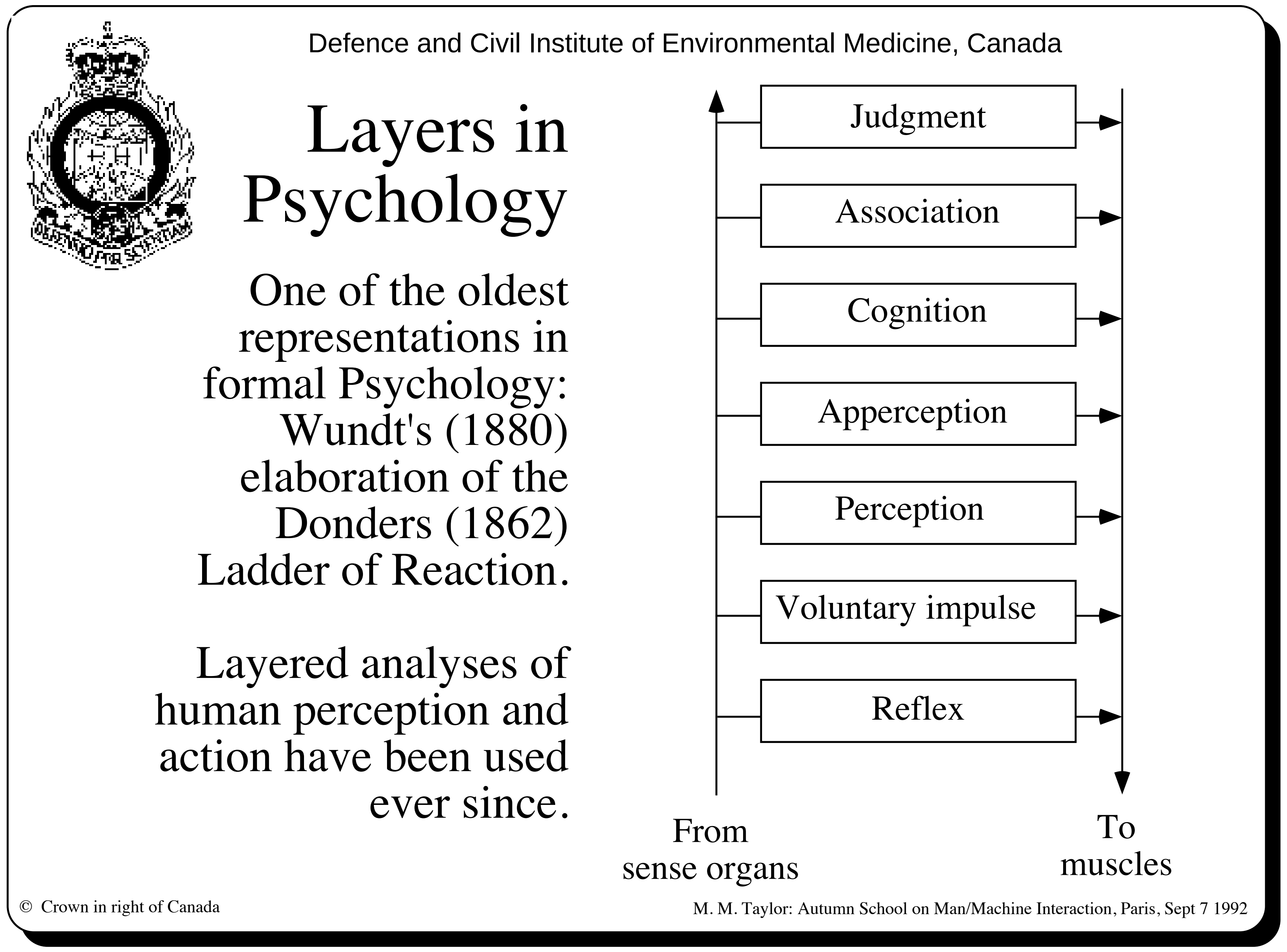
Layered Protocol (LP) Theory is an integrative framework that covers all aspects of communication.

LP Theory addresses WHY we communicate as we do, based on an evolutionary need to survive. It is based on goals and results, not on the overt content of messages.

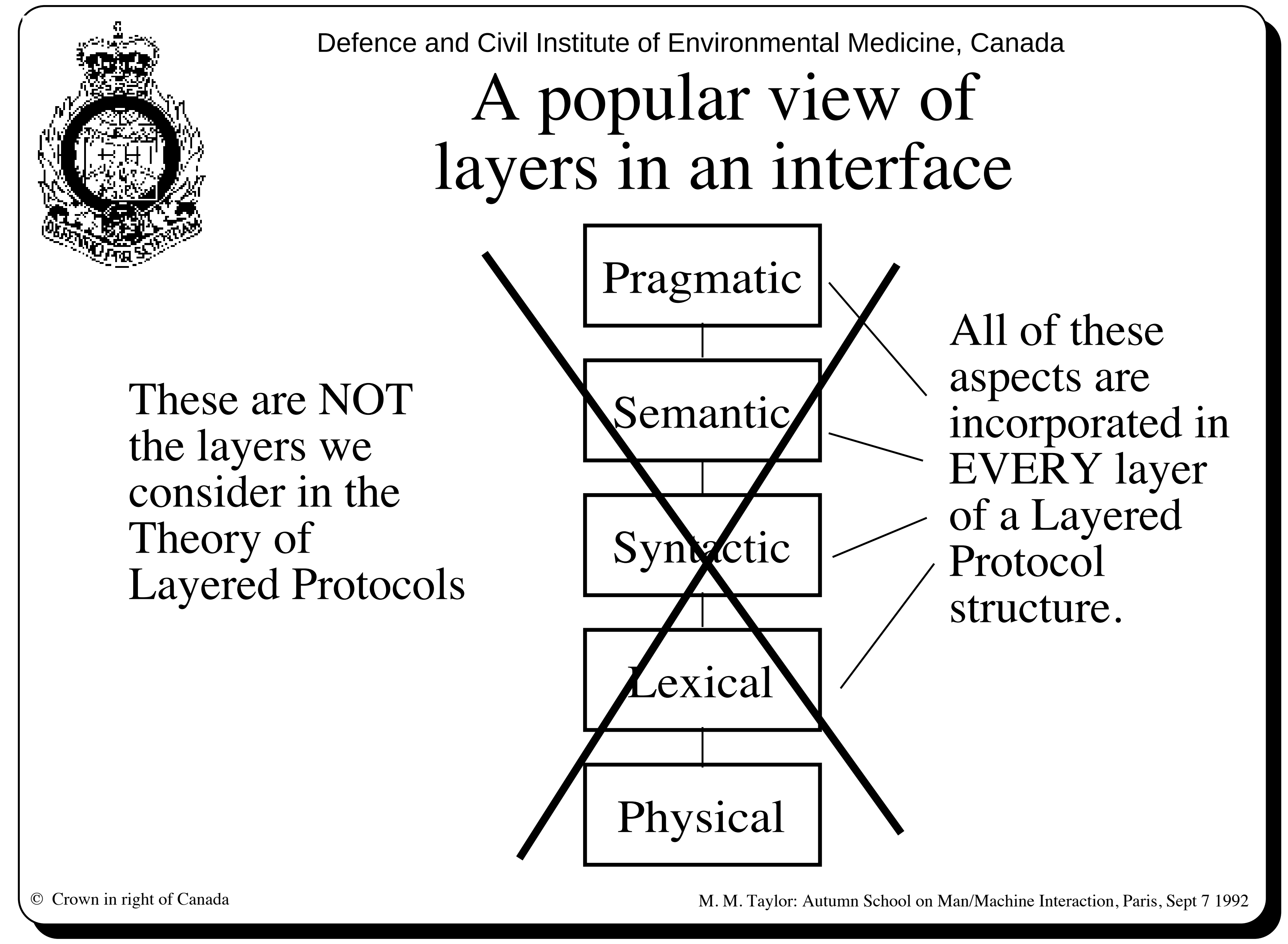
LP Theory addresses HOW we communicate as we do, based on the need for a communicator to determine whether its communications have succeeded. It is based on feedback control systems.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

1-16



1-17



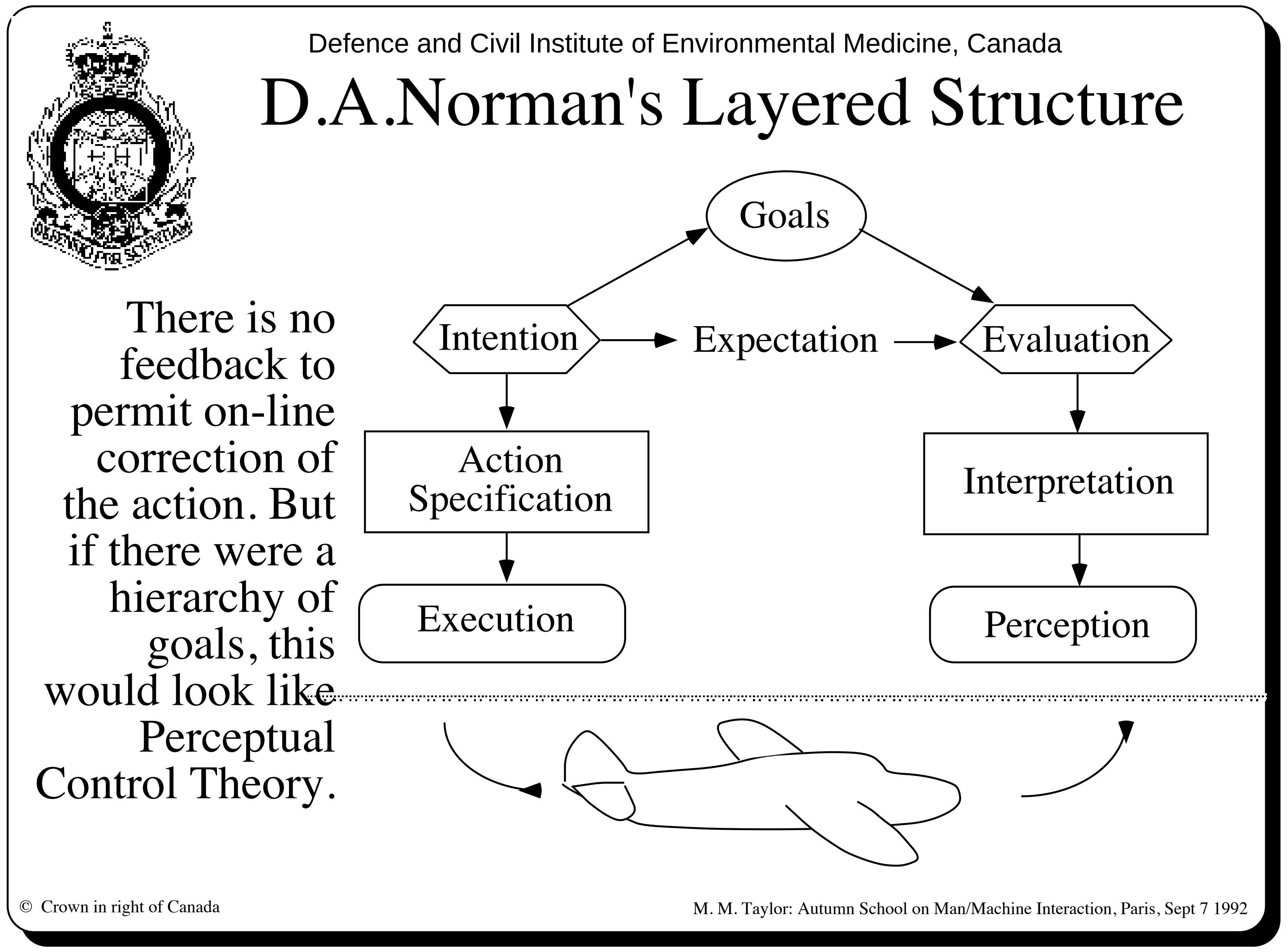
The idea that people work with layers of abstraction is as old as experimental Psychology. Donders (1862) and Wundt (1880) both thought that they could analyze the structure of perception and thought by measuring the reaction time of responses to events that occurred at different levels of abstraction. The lowest level might be the kind of reflex that occurs involuntarily in response to a tap on the knee or a puff of air to the eye, the highest a judgment such as whether a picture fairly represented the style of a particular artist (this example is invented, not proposed by Donders or Wundt).

Although these particular levels of abstraction and simple ladder structure are no longer used, the concept that people work with levels of abstraction in an important way is still very current. It is inherent in both Perceptual Control Theory and Layered Protocol Theory.

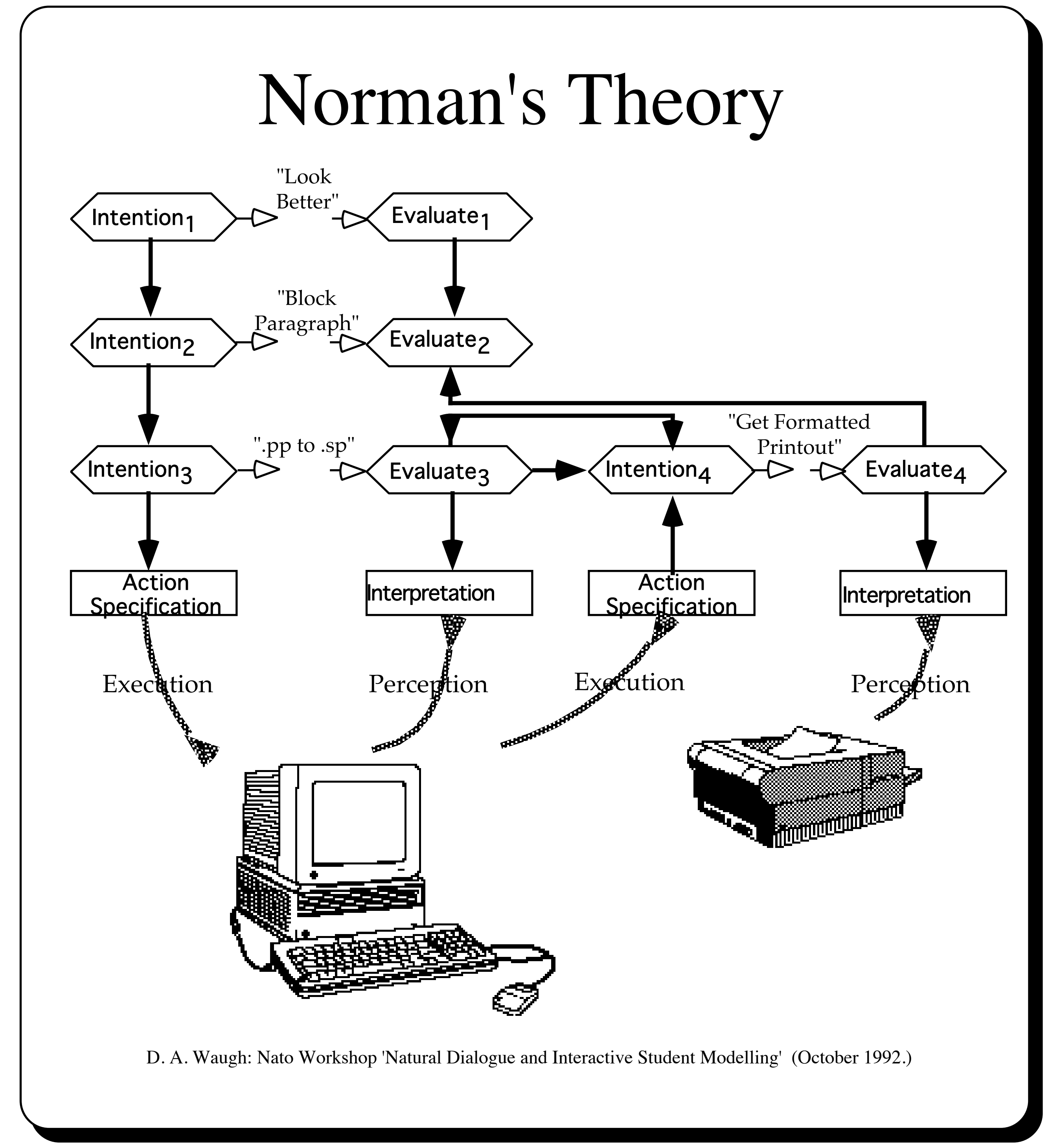
Most often when researchers in Human-Computer Interaction talk about layers in the interface, they assume that there is an incoming physical stream of events that are given meaning in a series of discrete stages of decoding. First the event stream is divided into tokens that are the lexical elements of a vocabulary of interaction. These tokens are then related according to the rules of some syntax, after which the syntactically correct forms are identified as having some semantic content, and then (possibly) associated with the pragmatic situation to take on a context-specific meaning. We think this approach is wrong.

There are other, more complex layered structures in the literature, but for the most part they are closely related to the one depicted. The layers of LP Theory are quite different; each layer combines all these aspects, but at different levels of abstraction.

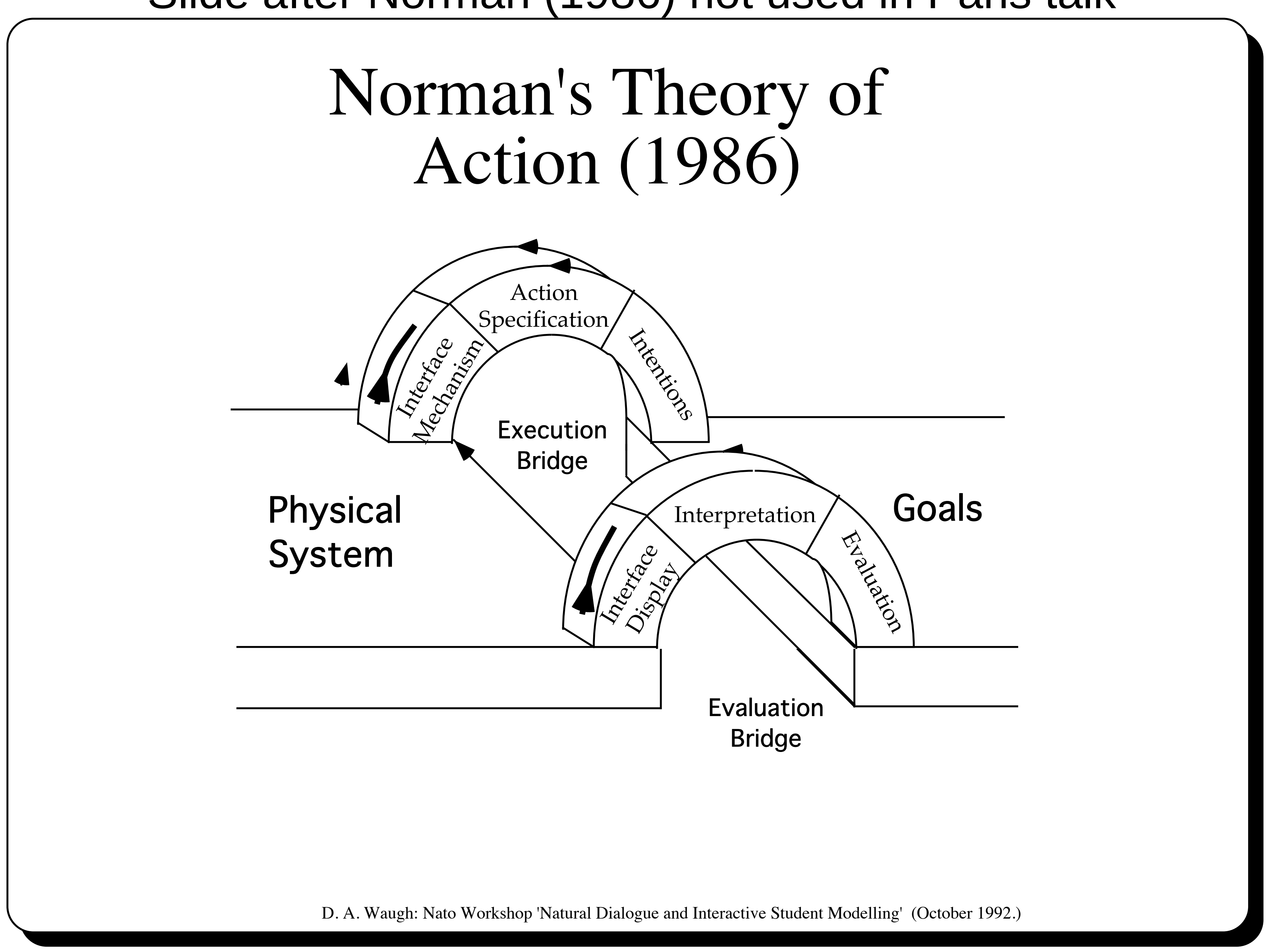
1-18



1-1 Slide after Norman (1986) not used in Paris talk



1-12 Slide after Norman (1986) not used in Paris talk



D.A.Norman (1986;*Cognitive Engineering*, in D.A.Norman & S.W.Draper (Eds) *User centered system design: New perspectives in human-computer interaction*. Hillsdale, N.J.; Erlbaum) has developed a theoretical structure quite like PCT. Action is initiated by a discrepancy between the actual state of the world and a goal state. The discrepancy sets up an intention to act, specifications for action are produced, the action is executed and its effects perceived, interpreted, and evaluated. Action specifications can be goals for lower-level actions, setting up a hierarchy of levels of behaviour.

The lower left picture shows the loop as a pair of encoding and decoding bridges between the internal desired states (Goals) and the actual states of the world (Physical System). The picture in the upper right illustrates both a hierarchy and a temporal sequence of goals and actions.

1-19

Defence and Civil Institute of Environmental Medicine, Canada

Layered Protocol Theory History:

1981-4: LP concepts developed as a technique for easing the design of Human-Computer Interfaces, and for the analysis of human communication.

1986: LP Theory re-founded on the basis of information theory and feedback control theory. LP seen as a necessity for "intelligent" real-time communication.

1991: LP found to be an instantiation of Perceptual Control Theory, itself a powerful framework theory for psychology.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

1-20

Defence and Civil Institute of Environmental Medicine, Canada

Basic Principle of Layered Protocols

All communication is the control of belief

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

1-21

Defence and Civil Institute of Environmental Medicine, Canada

Basic Principle of Layered Protocols

All communication is the control of belief

Why?

If I want you to have some information, I have to believe either that you have it or that you have not.

If I believe you have the information, I do nothing.

If I believe you do not have the information, I communicate, and continue to do so until I have attained a belief that you do.

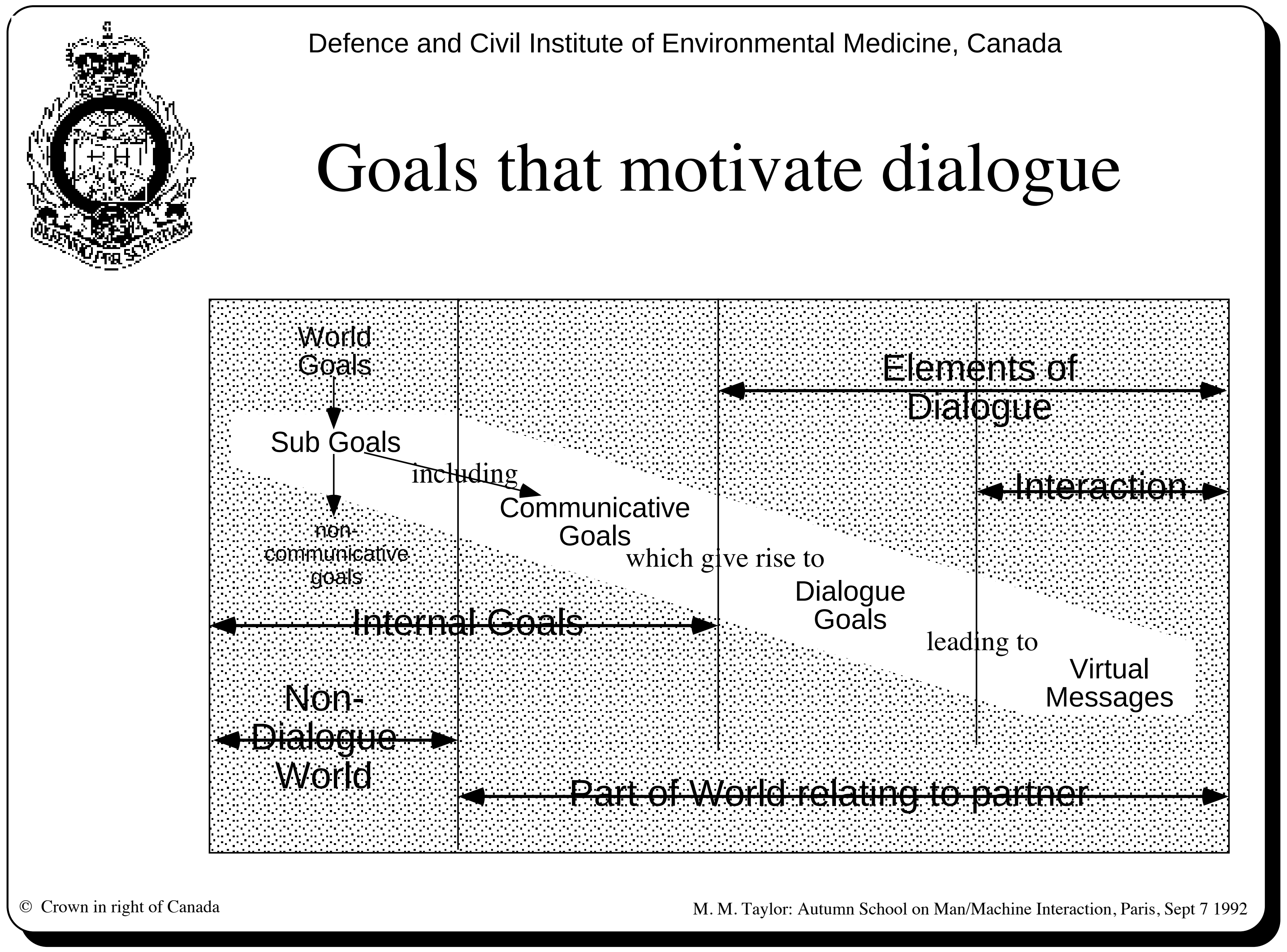
I am controlling my state of belief about your information.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

All that anyone can use as a guide to what to do is their own belief about the current state of the world and about ways to change those beliefs by acting on the world. In communication, those beliefs are about the communicative partner. If I want to be able to believe something about your beliefs that is different from what I now believe about them, I must do something. That something is the transmission of information to you that will alter your belief and, if you are being cooperative, you will in some way transmit information to me that allows me to adjust my belief. I can continue sending you information until I believe that your belief structure is the way I would like it to be (or until I give up trying).

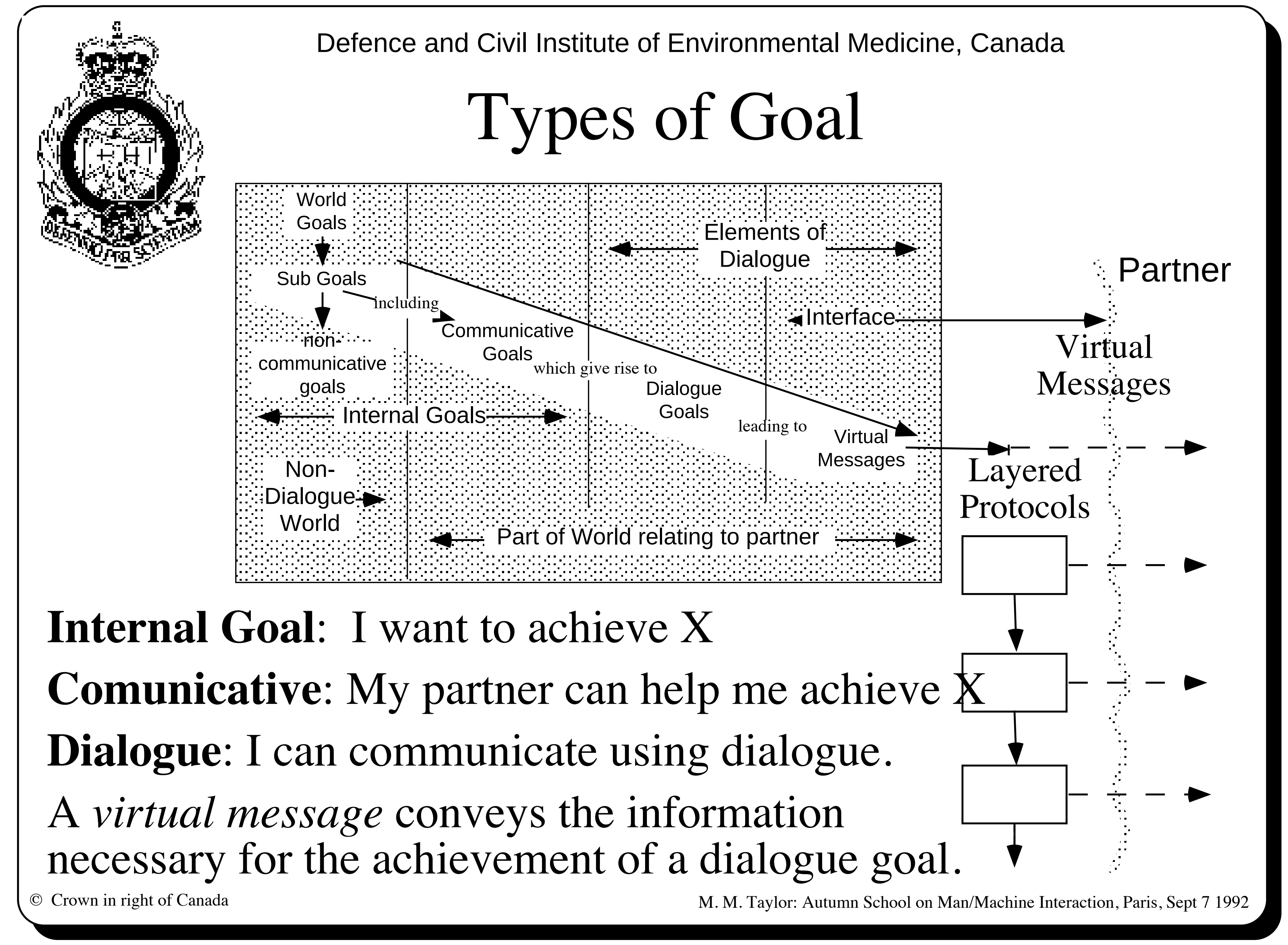
Some communication is done without feedback, in the expectation and hope that it will have the desired effect, but such communication does not represent interaction.

1-22



Communication is part of a larger world. **Communication is not normally the main task; it is in support of a task**. A person has a goal (to bring a perception to a desired state). The goal can be satisfied if a variety of subgoals are satisfied. Some of these subgoals may involve the participation of another person (or computer). To obtain this participation requires communication, so these are communicative goals. Not all communicative goals involve the cooperation that is part of dialogue, nor, indeed, the knowledge of the other party that communication is occurring. In Shakespear's *Othello*, for example, Iago drops a handkerchief so that Othello will think his wife to have been unfaithful. It is essential to Iago's goal that Othello not know that the communication is from Iago. But most communicative goals involve the cooperation of the partner, so they lead to dialogue goals, which both parties cooperate to satisfy.

1-23



A dialogue goal requires the passage of a virtual message between the parties. A virtual message is defined as whatever information must be passed from the originator to the recipient to allow the originator's belief about the recipient's beliefs to come to the desired state (i.e. to allow the originator to satisfy the communicative goal). A virtual message may be very complex, such as an entire philosophical discourse. In Human-Computer Interaction, it is usually much simpler, such as the specification of a drawing, or of a workspace layout with some complex functionality.

In order to transmit a virtual message, it must be transformed into other virtual messages at successively lower levels of abstraction, until finally the message has a physical form that can bridge the gap between the partners. Those transformations are performed by layered protocols.

END OF PART ONE

Defence and Civil Institute of Environmental Medicine, Canada

Part 2

Introduction to Perceptual Control Theory

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Part 2 of the talk provides a preliminary look at Perceptual Control Theory, which is a fundamental basis for understanding the interactions of living systems (including people) with the world they live in. Everything else in this tutorial should be seen as it relates to Perceptual Control.

The concept of "Perception" should be taken more broadly than perhaps is common in psychology. In Perceptual Control Theory it applies to any useful function of sensory input, or, sometimes elements of the imagination, which may mimic functions of sensory input. "Perception" includes perception of the meanings of words, as well as of the states of objects in the world, and of the most abstruse concepts. A perception need not be conscious, and most perceptions considered in Perceptual Control Theory are not. The tension in a muscle is perceived and controlled during an action, but is seldom if ever brought into consciousness.

Defence and Civil Institute of Environmental Medicine, Canada

Basic Principle of Psychology

All behaviour is the control of perception

© Crown in right of Canada

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

All behaviour is the control of perception is a motto that should be burned into the brain. Behaviour is what one does intentionally. If I pick up a glass to drink, that is behaviour, but if I knock the glass over in trying to pick it up, the knocking over is not considered as behaviour. It is a byproduct of my actions. The motto is very important, because any actions that do not participate in the control of some perception are wasted energy. There may well be such actions, but they are always accidental by-products that are not part of what the organism (or person) is "doing." They would not be recognized by the actor as being part of the behaviour. An outside observer looking at a person's actions cannot tell what a person is actually "doing," i.e. what perceptions the person is controlling.

All behaviour is the control of perception is critical in deciding what should be presented on a computer's display surfaces or other output devices.

2-2

Defence and Civil Institute of Environmental Medicine, Canada

Basic Principle of Psychology

All behaviour is the control of perception. Why?

If you perceive that something is as you want it to be, you do nothing about it.

If it is not as you would wish, you act so that you can perceive it to be more like your wishes

The world changes for many reasons, altering what you perceive

There are many ways to alter the world so that your perception becomes more like your wishes.

Your actions control your perceptions, not the other way around.

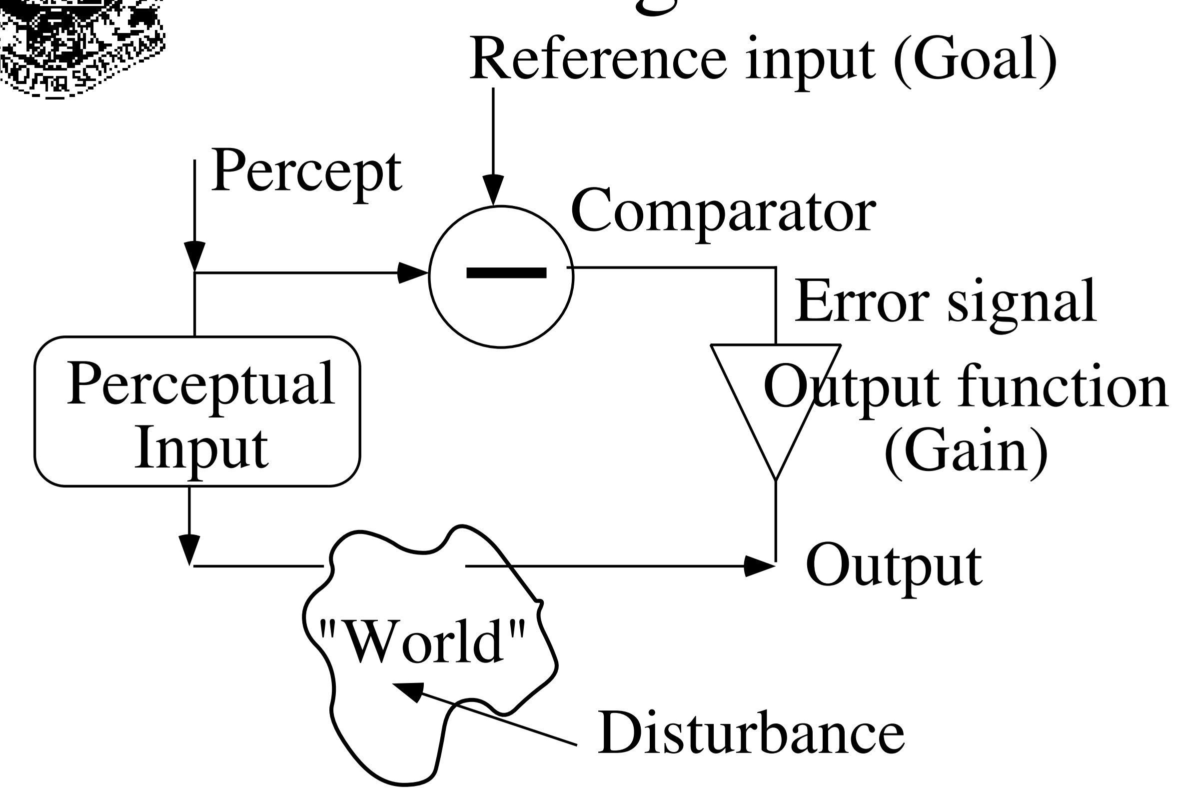
M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The fundamental construct of a control system is the feedback loop. Some aspect of the state of the world is sensed (the percept of the control system) and compared with a reference signal that specifies the currently desired value of the percept. The difference is the error, which is amplified to create the output of the control system. This output has some effect on the world that changes the percept. If that change is such as to bring the percept nearer the reference, thus reducing the error, the feedback is negative and there is control. The amount by which a unit error would affect the percept if the comparator connection were broken is called the loop gain. Since the loop gain must be negative if there is to be control, we often ignore the sign in quoting its magnitude, unless it is important to distinguish in a particular situation whether the feedback is negative (control) or positive (exponentially increasing error).

2-3

Defence and Civil Institute of Environmental Medicine, Canada

Feedback: Positive and Negative



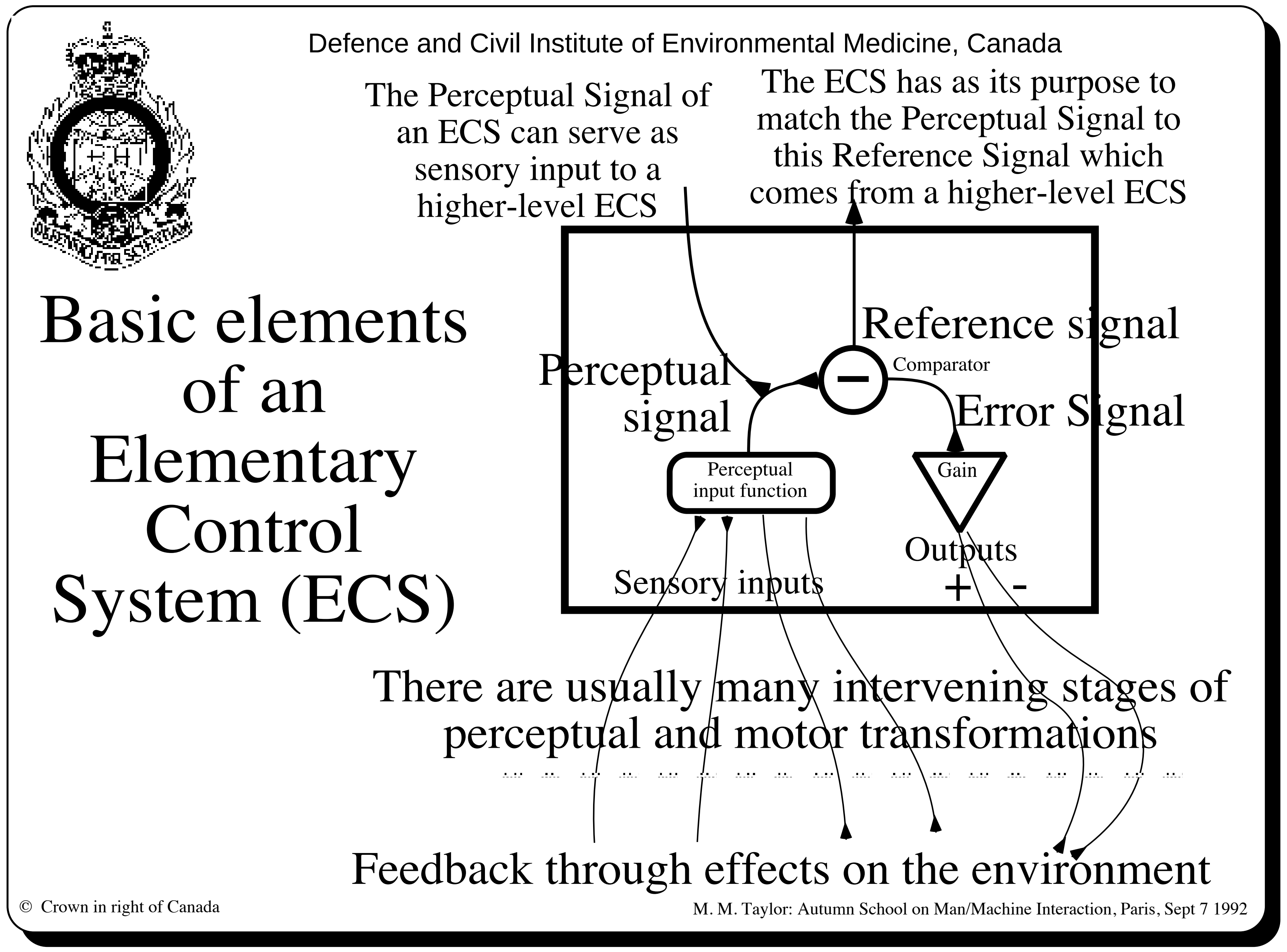
When the disturbance makes the percept get smaller, causing error, what happens if the effect of the output is to reduce further the value of the percept? The error increases, and the percept gets even lower. This is positive feedback, and is not wanted in a control system. If the effect of the output is to oppose the disturbance, the feedback is negative, and there is control.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Typically, in an effective control system, the negative loop gain is much greater than unity. A control system with high gain is stiff, and strongly resists changes in its percept except insofar as they follow changes in the reference signal. A control system with lower gain is softer, and less resistant to disturbances that change its percept. Sometimes we use the term "insistence" rather than gain, because the psychological implications of "insistence" are clearer than those of "gain."

The "output" of the control system are the actions that affect the world in such a way as to counter the disturbance. If the gain is high, this means that the perceived variable does not change. The disturbance itself does not contribute to the percept, because it is countered by the actions.

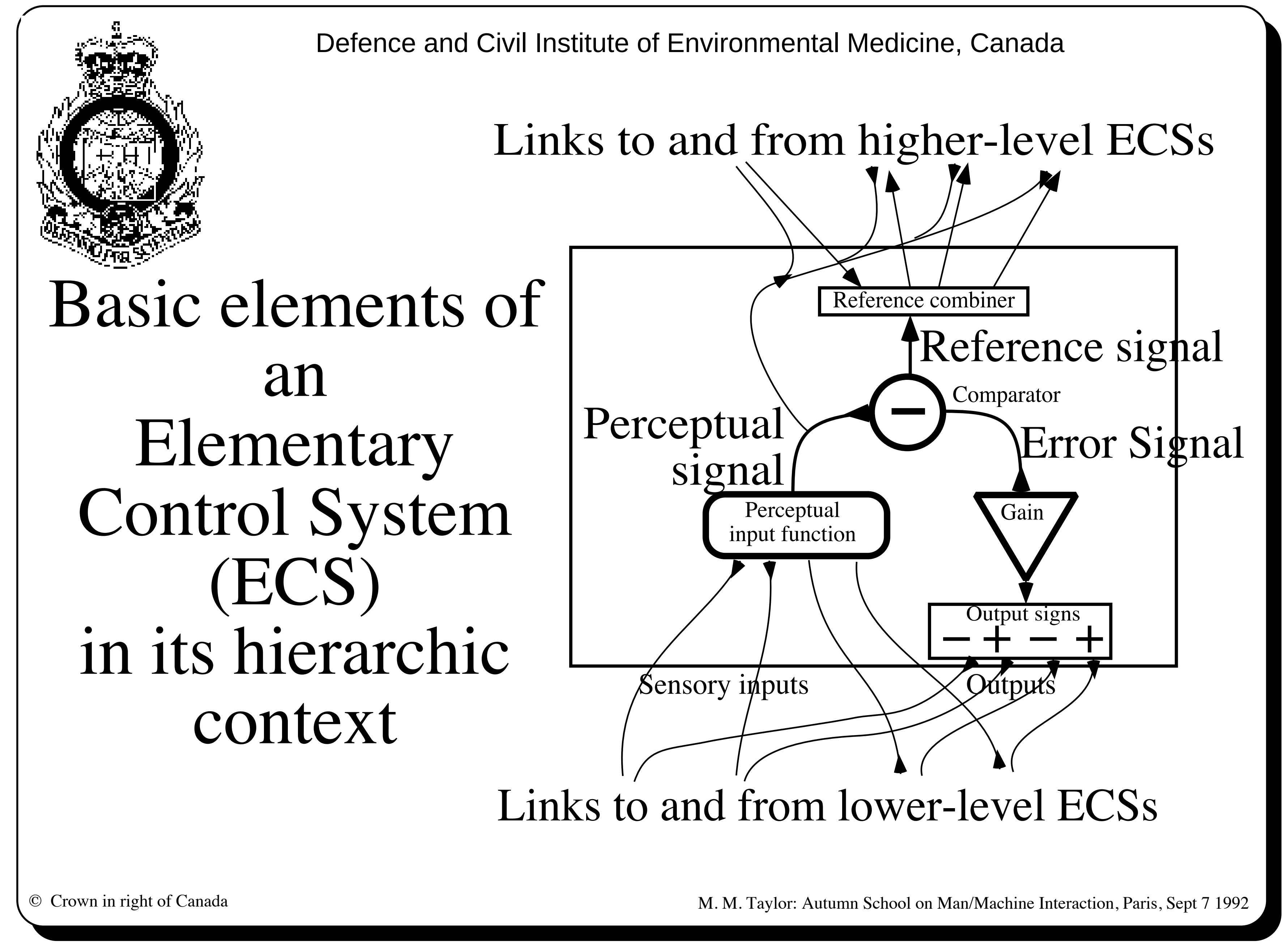
2-4



The basic unit of Perceptual Control Theory is an Elementary Control System (ECS). A control hierarchy normally consists of several interconnected ECSs. Each ECS has a set of main components, in addition to possible minor ones not incorporated in these figures. The main components are (1) a perceptual input function that combines possibly many sensory inputs into a single perceptual signal, (2) a comparator that compares the perceptual signal with a reference signal supplied from outside the ECS and provides the difference as an error signal, (3) an output function that transforms the error signal into an output signal—typically the output function is an integrating amplifier.

Reference signals normally are produced by combining outputs that come from higher ECSs, and the perceptual signals are output to become the sensory inputs of higher ECSs.

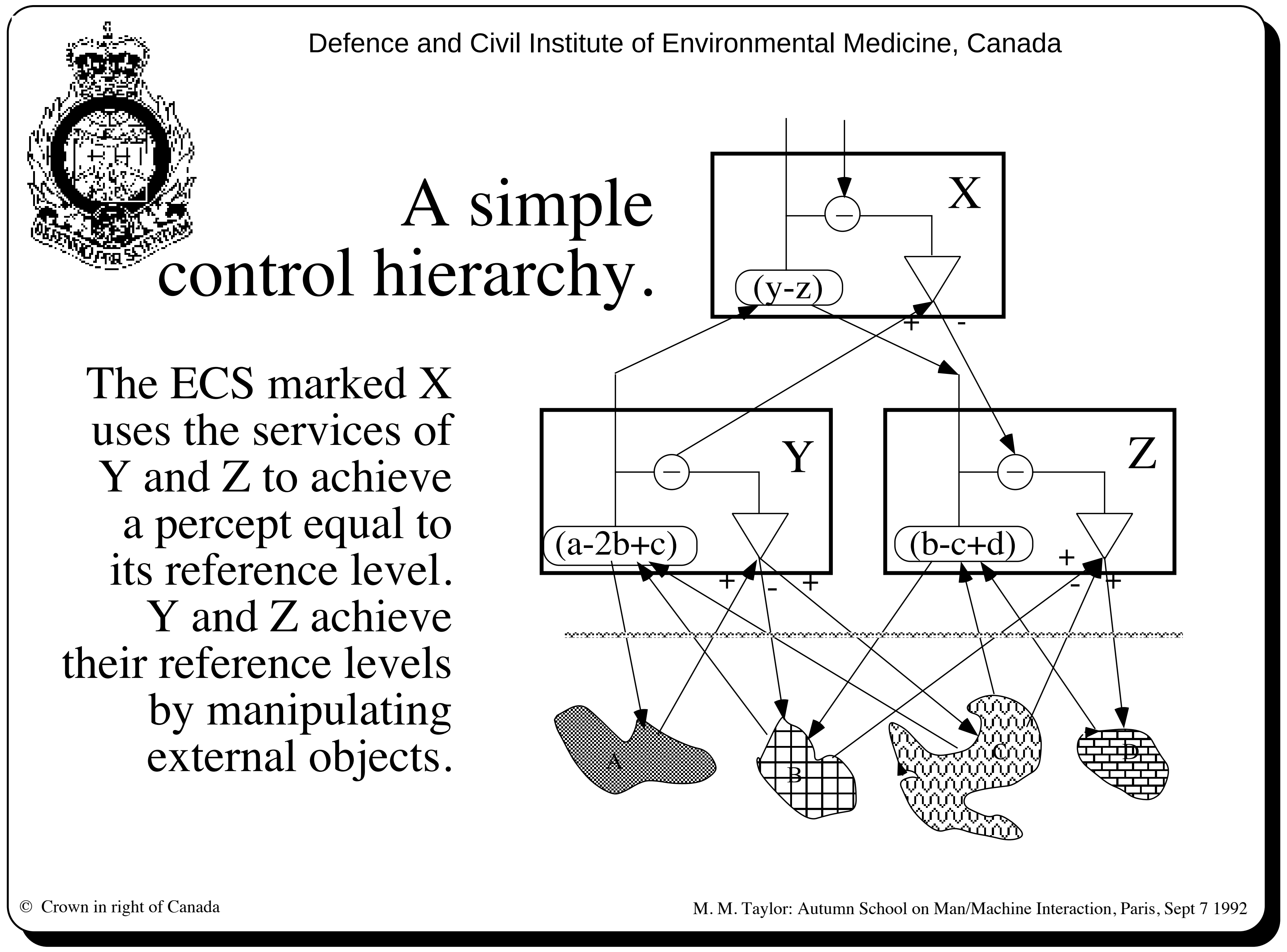
2-5



An ECS has a reference input function that combines the outputs from higher ECSs to produce its reference signal. A positive output from the output function of the ECS may be distributed as positive to some lower ECSs and as negative to others. In a well constructed network of ECSs, each output eventually results in a change in the perceptual signal in the direction that reduces the error signal. We will not consider in this talk how the correct linkages are formed.

Very often, in simulations at least, reference signals to an ECS come from the same places to which the perceptual signal is sent. One can conceive this relation as that a higher-level ECS requests the lower to deliver a particular magnitude of perceptual signal. In a more realistic (live) system, there may be some reciprocity, but it will not be so tight. The outputs of an ECS will affect possibly many other layers of ECSs and eventually the world, to have some effect on its own sensory inputs and hence on its perceptual signal.

2-6



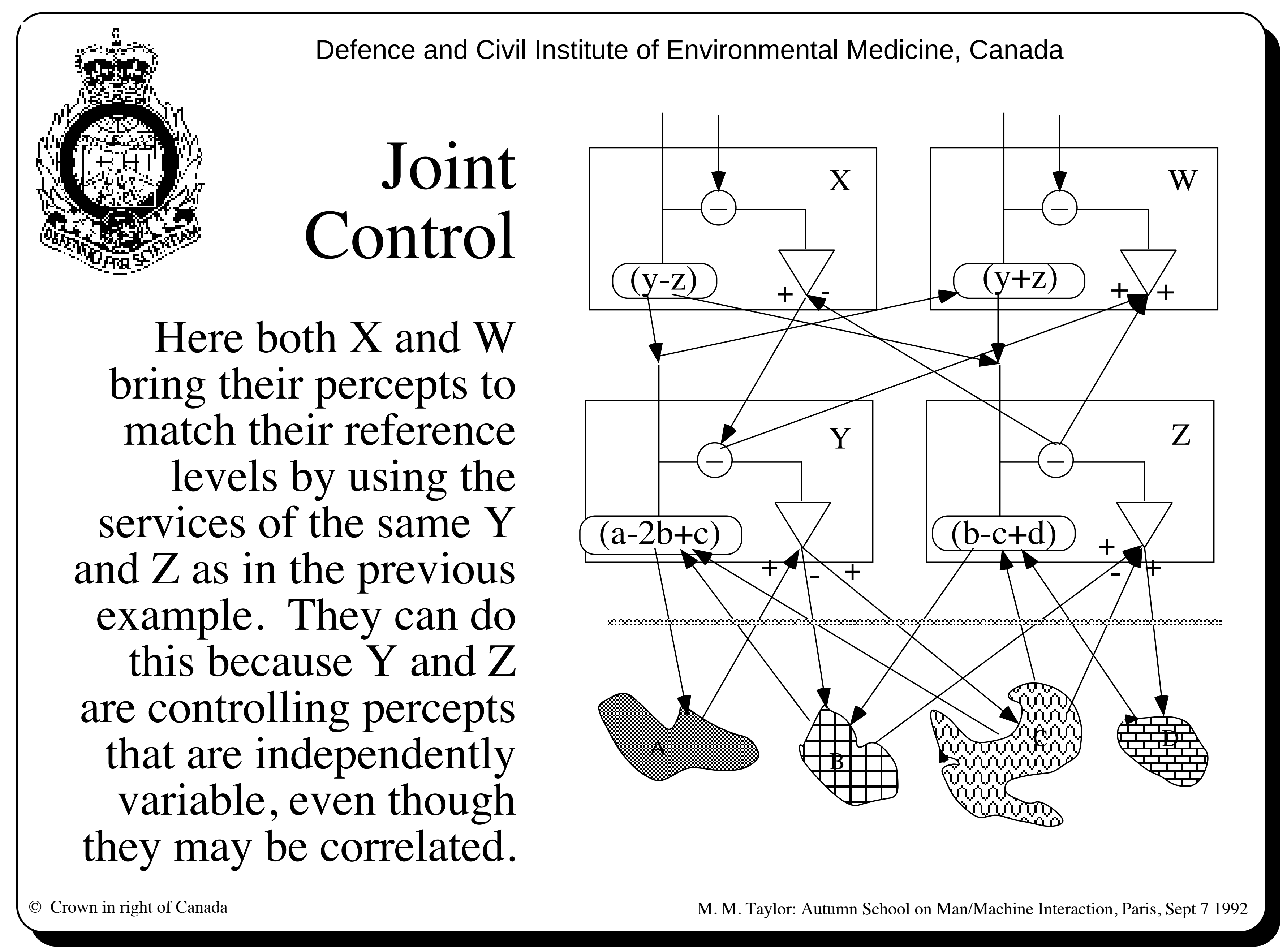
It may be helpful to trace out the flow of control in a simple two-level control network. The blobs marked A through D represent aspects of the world that can be affected by the outputs of ECSs Y and Z, and that provide sensory inputs a, b, c, and d to those ECSs. Y and Z in this example have perceptual input functions that can be described algebraically as y(a - 2b + c) and z(b-c+d) respectively.

Both Y and Z are provided with their reference signals by the output of ECS X, which goes to Y in the positive sense and to Z in the negative sense. In turn, Y and Z provide X with its sensory inputs, which are combined by the perceptual input function x(y-z). The reference signal for X comes from some external agency. X, being at a higher level, is assumed to react more slowly than Y or Z.

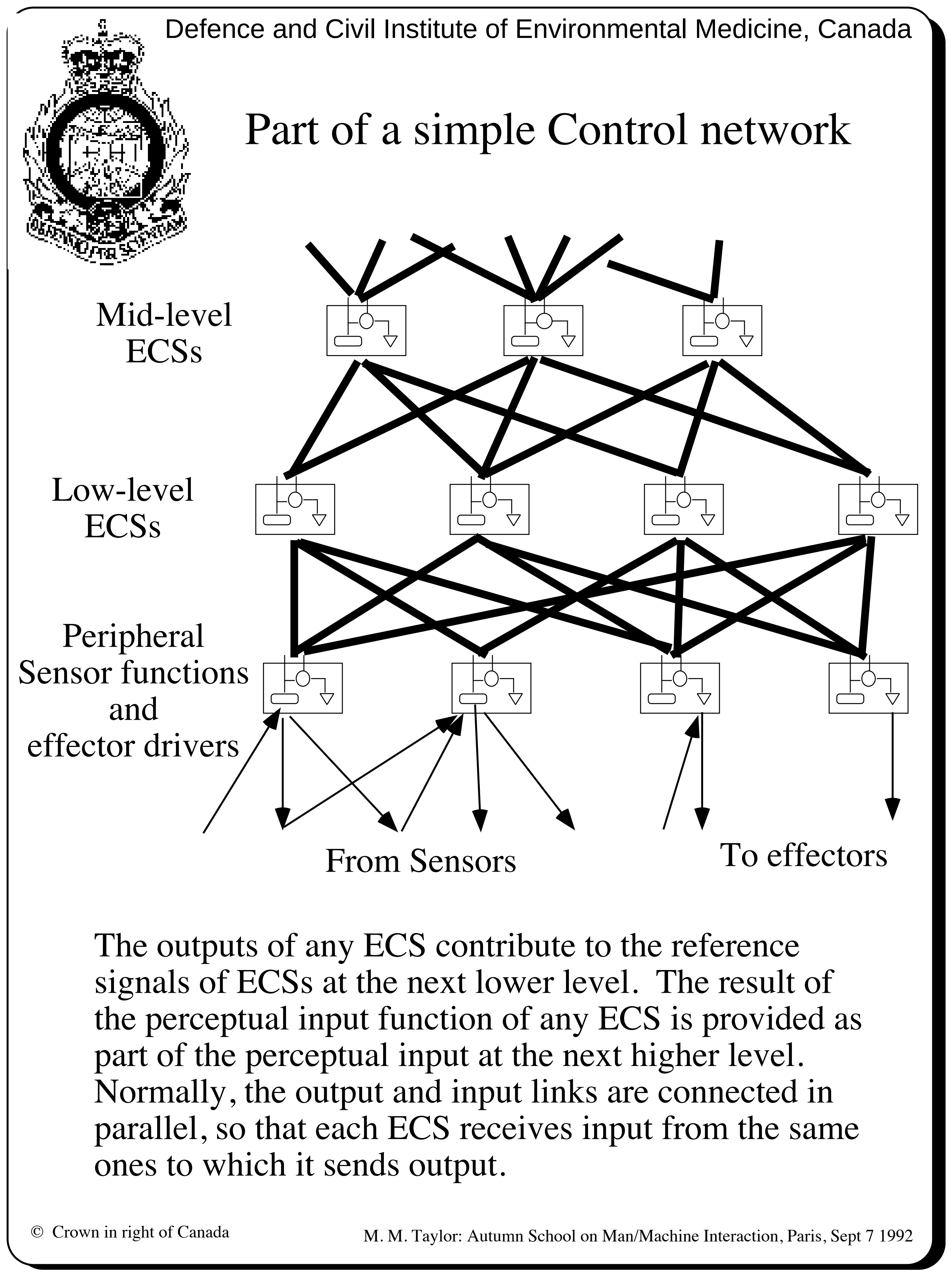
Now trace what happens if some external disturbancemoves, say D, so as to increase d. The perceptual signal of Z, which is a function of (b–c+d), will increase, causing error in Z, which produces output that tends to decrease b and d but to increase c. The effect is to resist the disturbance in d, but only to an extent that depends on how hard it is for Z's output to alter b and c relative to d. Z's effects on B and C may alter Y's perceptual signal, causing it to produce corrective output affecting a, b, and c, until its perceptual signal again matches its reference. The disturbance that was applied to D affected all four of the aspects of the world affected by the actions of Y and Z. But after all is done, Y and Z will still be perceiving what their reference signals direct, and there will be no lasting effect on X.

Now consider the second figure, and trace what might happen if, say, the reference signal for the new ECS, W, is altered. W will produce outputs that affect the reference signals and hence the perceptual signals of Y and Z, and these may well affect X, producing error. But the difference of Y and Z can be changed independently of the sum, so both X and W can be satisfied at the same time. Their feedback loops interact, but are not in conflict.

2-7

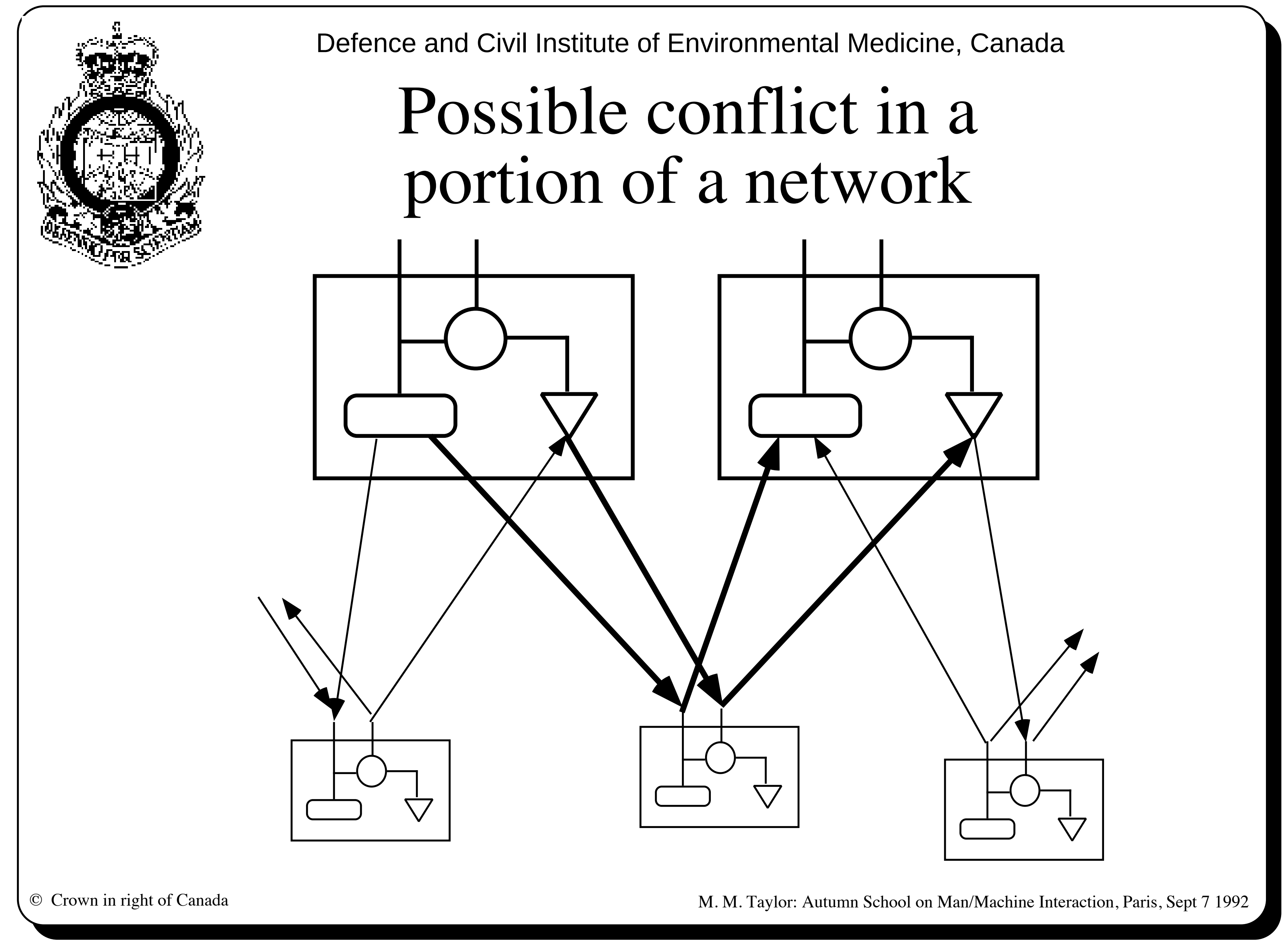


2-8



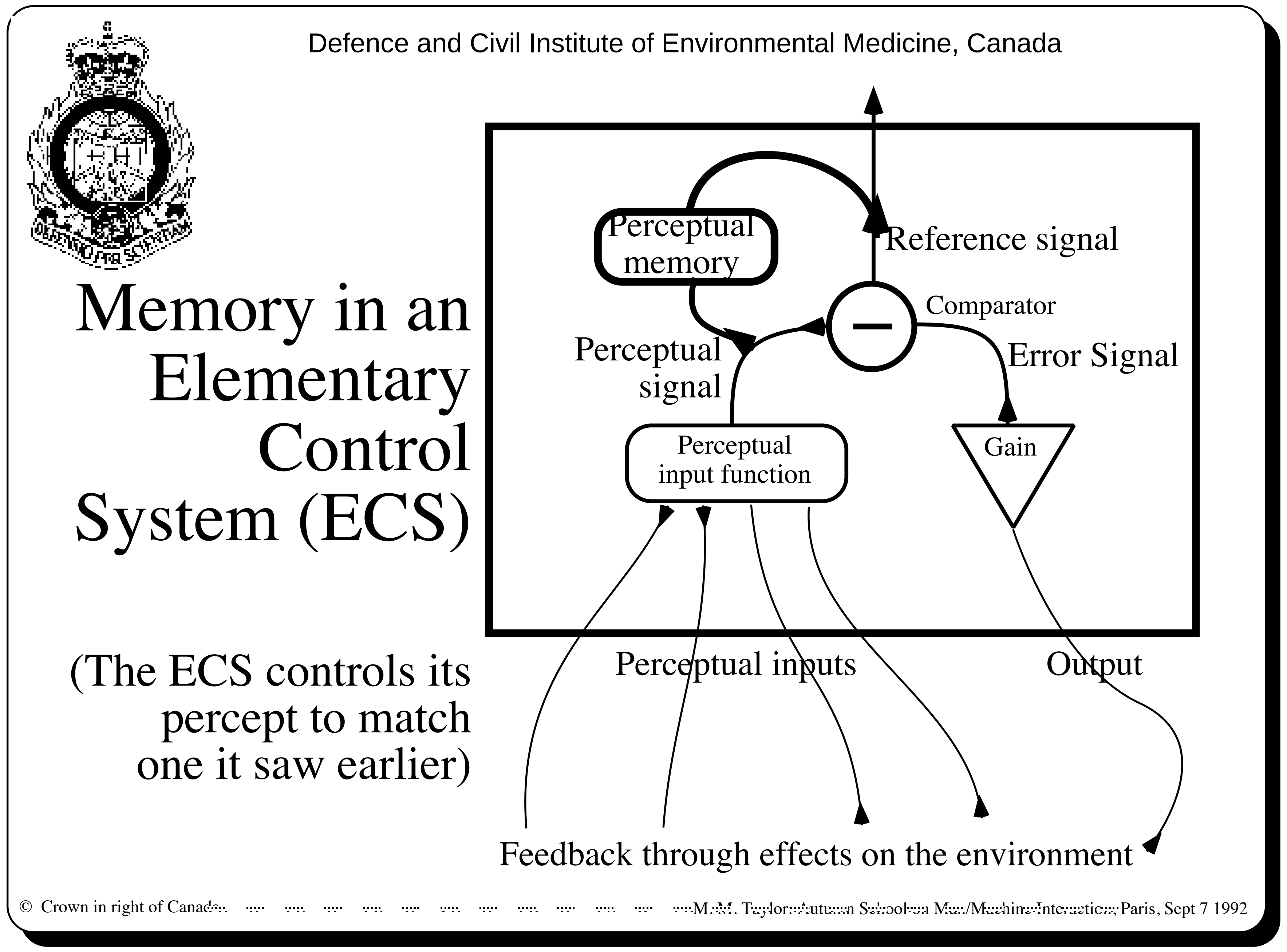
ECSs combine into a network of possibly many levels. At the lowest level, the ECSs are degenerate, some accepting sensory input for combining and passing on to higher levels, some taking the multiple reference inputs and sending output directly to effectors that operate on the outer world. The whole arrangement is reminiscent of a multilayer perceptron, except that there are parallel upgoing and downgoing signal paths rather than a one-way path. In fact, **if the perceptual input functions have a the form of a weighted summation followed by a nonlinearity, the sensory-perceptual part of the structure is exactly a multilayer perceptron**.

2-9

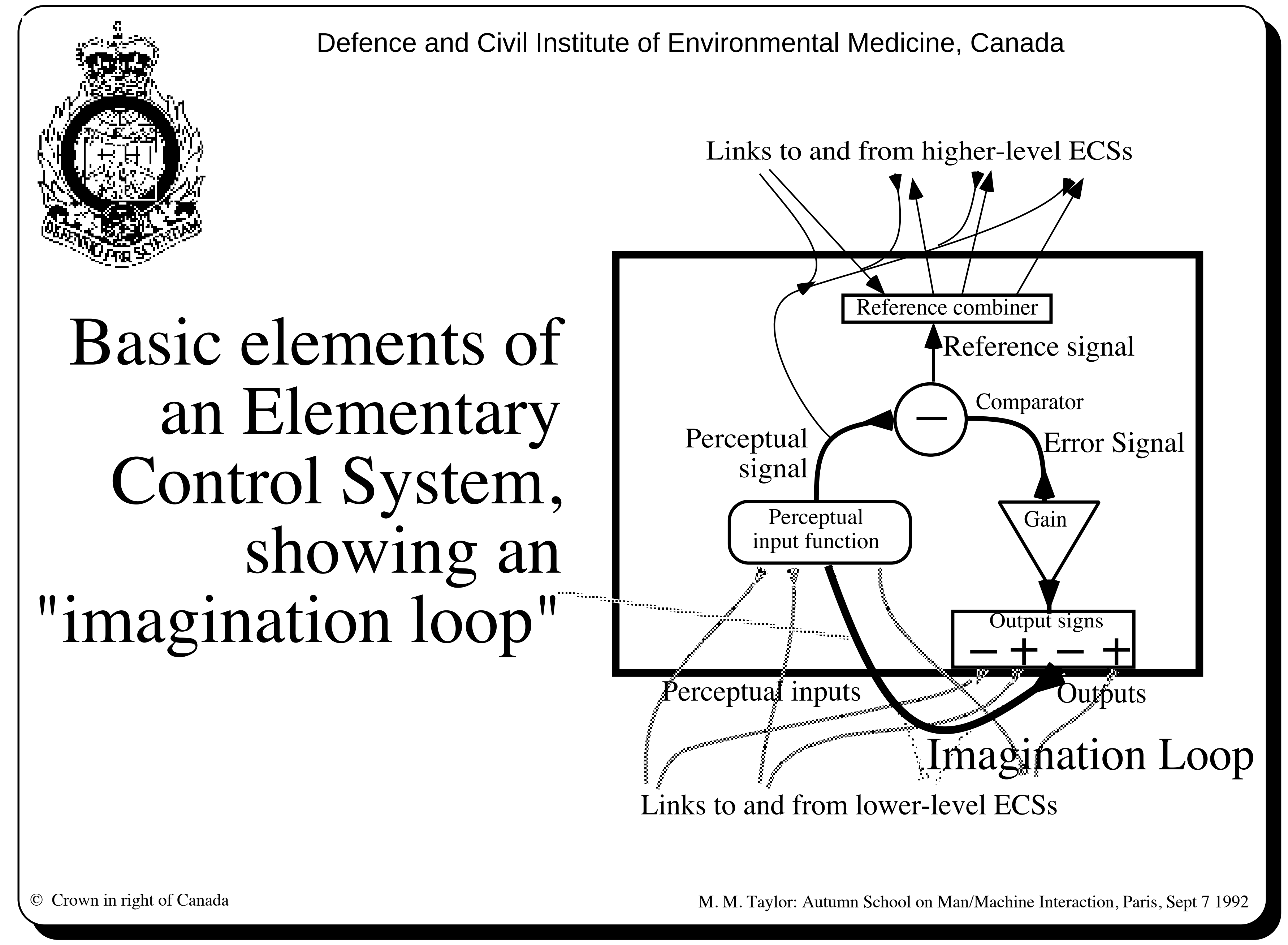


When two (or more) ECSs attempt to control perceptual signals that depend on a common pathway, there is the opportunity for conflict. In this figure, the upper two ECSs both use the central lower ECS as part of their way of controlling their perceptual signals. This situation may, but need not, lead to conflict. If their requirements can be met by the ECSs at the lower left and right, then the central lower ECS can come to a state that provides an acceptable perceptual signal to the higher ones, when combined with the signals from the other two. But if the ones at the lower left and right are in some way inhibited, then it is not possible for both the higher ones to bring their error signals to zero simultaneously, and at least one of the higher ECSs (probably both) will continuously provide output that tries to move the central lower one. If the output functions include error integration, as is common, then the "force" applied by each to the central lower ECS will continually increase without limit.

2-10



2-11



An ECS may have elements other than the main ones mentioned earlier; in particular, it may have memory and imagination. Memory is seen as a record of a perceptual signal achieved at some earlier time, which can be used as a reference signal so that the same percept can be achieved again. We do not refer much to memory in the remainder of this talk, and include it here more for completeness than for immediate use. The connection shown here is simplistic, and a full discussion would be complex and unnecessary here.

Both memory and imagination can, in principle, be used at the same time as pathways that feed back through the real world, but they can also be used as substitutes for real world sensation and action.

Imagination is a way for the ECS to determine the probable effect of an output signal without actually acting on the outer world. The perceptual signal is taken directly from the output, rather than being provided by a lower ECS for which the output would have provided a reference signal. It is as if a wine expert, instead of sipping from a glass of red liquid, said to himself "What would the taste be if I drank that Chateau Magritte '97?"

Imagination is a very important construct in dealing with attentional focus and alerting functions. It allows one to control certain perceptions of the real world while controlling other, perhaps incompatible, perceptions simultaneously without the constraints imposed by the world.

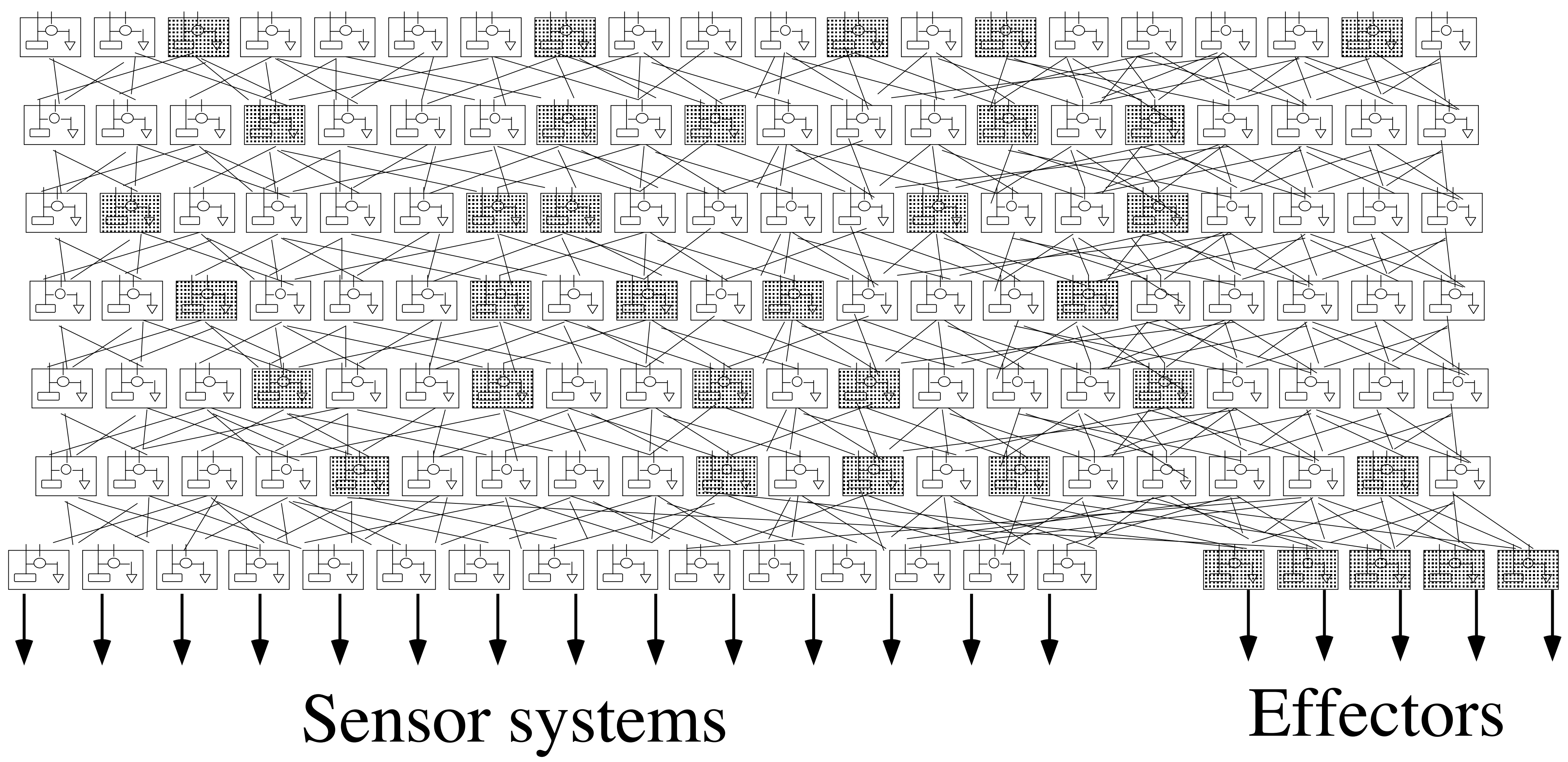
Defence and Civil Institute of Environmental Medicine, Canada

A control net with more sensor than effector degrees of freedom

At each level, no more ECSs can be in effective control of their percepts than there are effector degrees of freedom.

An ECS in effective control of its percept

An ECS unable to control its percept except through imagination

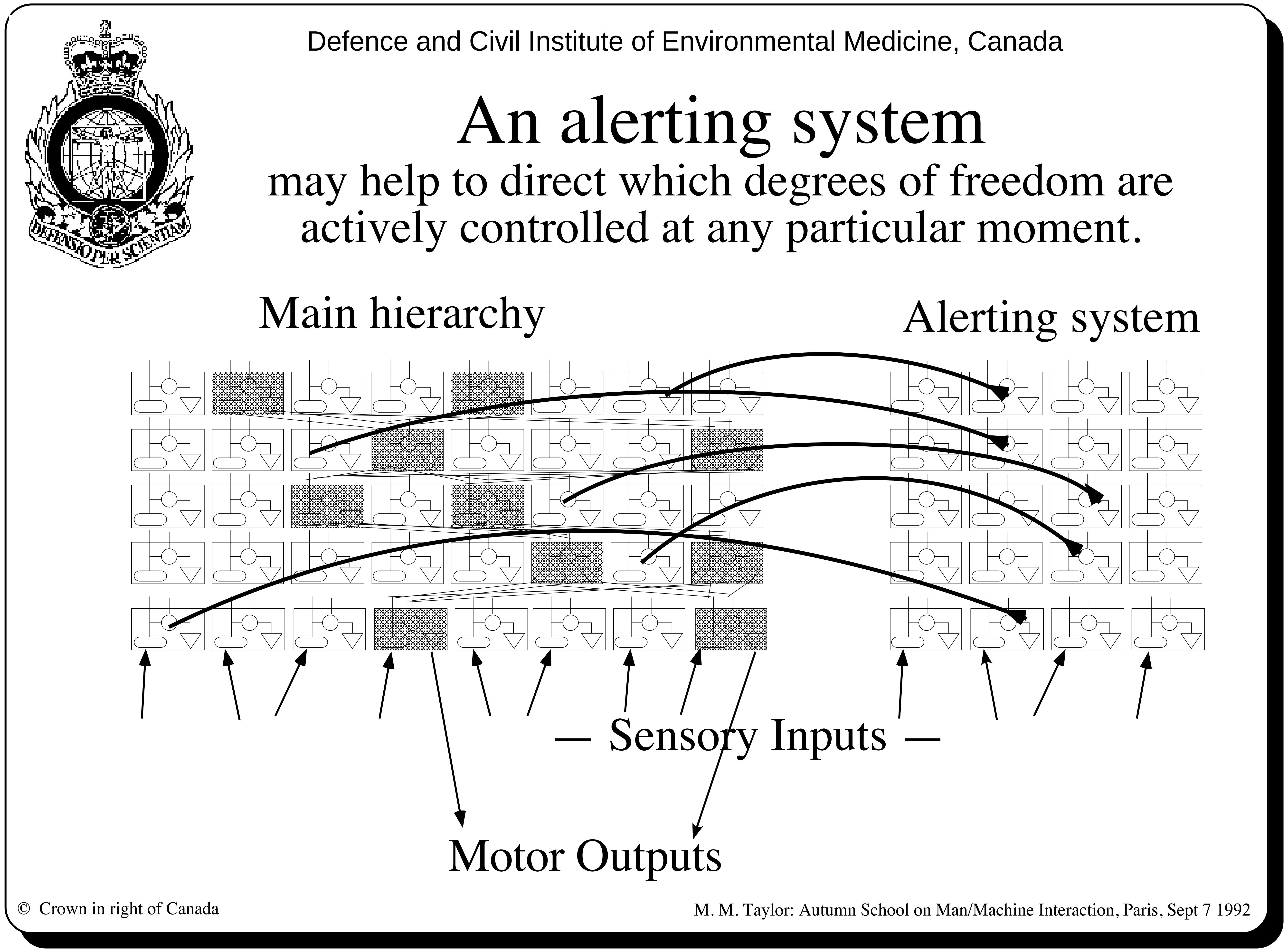


M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

A mobile control system, such as a human, has vastly more degrees of freedom for sensing than for output. A back-of-the-envelope calculation suggests that a human might have as many as 125 degrees of freedom for output, counting each main joint separately (even though many can move only in coordination) and allowing some for facial expression and control of the speech organs. At the same time, there are about a million fibres in each optic nerve, 30,000 in each auditory nerve, and a large number of tactile and other sensors. Even allowing for fixed recombination of these inputs based on the stable statistics of the environment, there are orders of magnitude more sensory degrees of freedom than effector degrees of freedom, and this ratio is enhanced when one considers the rates at which each can vary—tens of Hz for the sensors, single-digit Hz or less for the effectors.

The numbers for sensor and effector degrees of freedom, as well as simple introspection, show that there are at any moment many percepts that could be being controlled but are not. At any level of the hierarchy, there can, in principle, be no greater number of independently controlled percepts than there are degrees of freedom for output. But a much greater number could come under control at different times, though not simultaneously. ECSs that would control for percepts not currently under control (if they exist) can be controlling only through imagination. They could be tracking their percepts, determining whether the world, by chance, was maintaining them near their reference levels.

There must be some mechanism whereby the potential control of ECSs that at any moment are only imagining becomes actual, taking control from some ECSs that had been in actual control. We characterize this mechanism as an "alerting system," but do not argue for any particular way the alerting system works. Several different kinds of alerting mechanism may all be used. The next slide shows one kind of possibility.



If not all the ECSs can be in control of their percepts at the same time then sometimes control must be passed from one ECS to another at the same level. There must be some kind of mechanism that determines that a presently uncontrolled percept is worth bringing under control. Perhaps some percept has gone beyond "tolerable" bounds, or perhaps some "possibly dangerous" stimulus pattern has occurred. In the first case, the out-of-limits percept must be brought directly under control, whereas in the second, the percepts to be controlled need not have anything obvious to do with the "danger" pattern. The figure shows an explicit set of pattern-recognition systems dedicated to the alerting function, but there are many other possible ways to change the locus of control.

For an example of the first situation, imagine driving (carelessly) along an empty wide road, and taking the hands off the steering wheel in order to unwrap a sandwich. For a while, the car proceeds satisfactorily along the road, even though it is not under control. But eventually it will veer to one side further than the driver is willing to allow, and the unwrapping must stop while the controlled percept of car position in the lane is brought within tolerance limits.

For an example of the second situation, consider superstitious behaviour. The actions are performed so that the "bad" perception will go away. Superstitious actions may even work, especially if someone else is using them as part of their own controlled perceptions. For a more direct example, consider the value of wearing a specific type of clothing or uttering a cryptic password so as to avoid getting shot in a war.

2-14

Defence and Civil Institute of Environmental Medicine, Canada

Some attributes of a PCT or Layered Protocol hierarchy

Lower levels in the hierarchy act quicker than higher levels.

Lower levels support many kinds of high-level ECSs, and are therefore less task-specific than higher levels

The actions of higher levels are more programmatic and complex than those of lower levels.

Higher levels are more task-specific than lower levels

Higher levels are more likely than lower to use models

2-15

Defence and Civil Institute of Environmental Medicine, Canada

Aphorisms

The good is the enemy of the best

If it ain't broke, don't fix it.

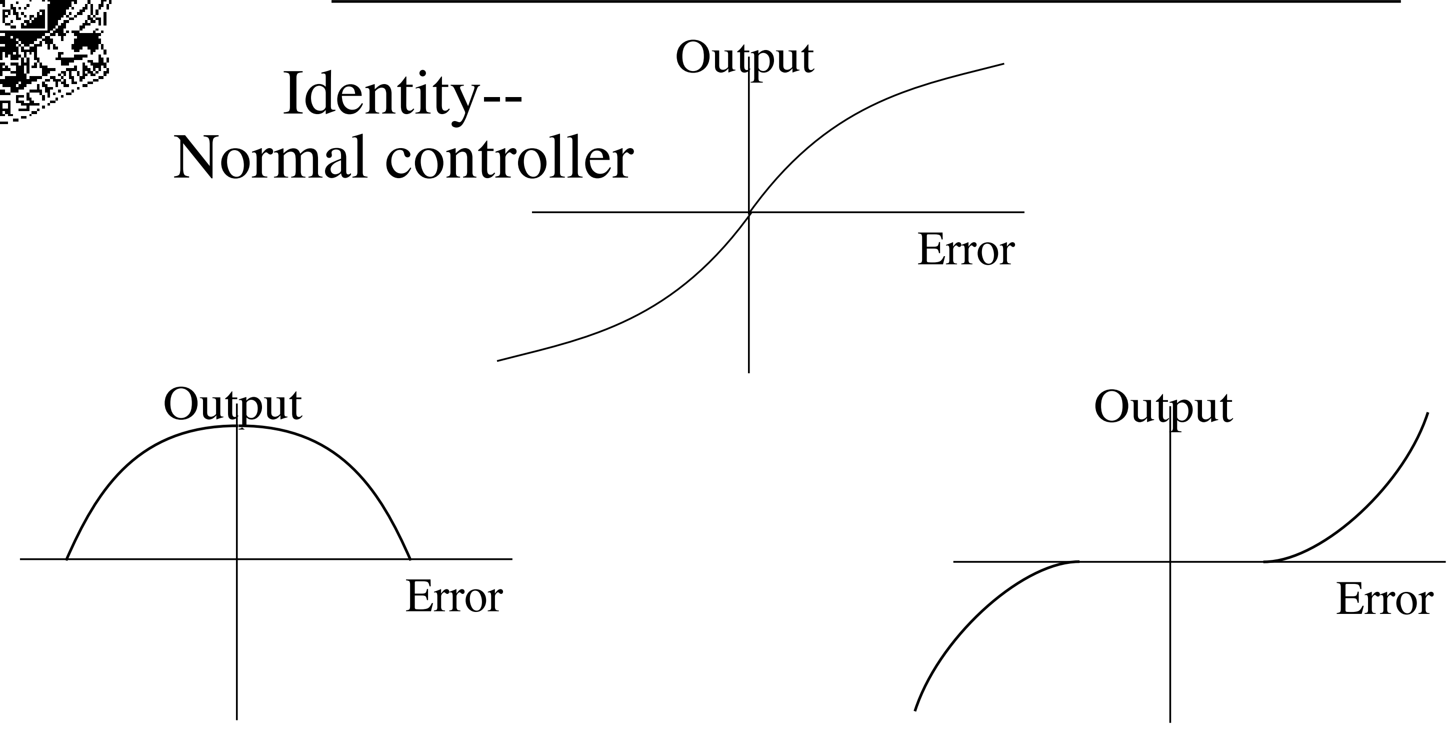
Are these in conflict?

2-16

Defence and Civil Institute of Environmental Medicine, Canada

Three essentially different types of gain function for an ECS

Identity--
Normal controller



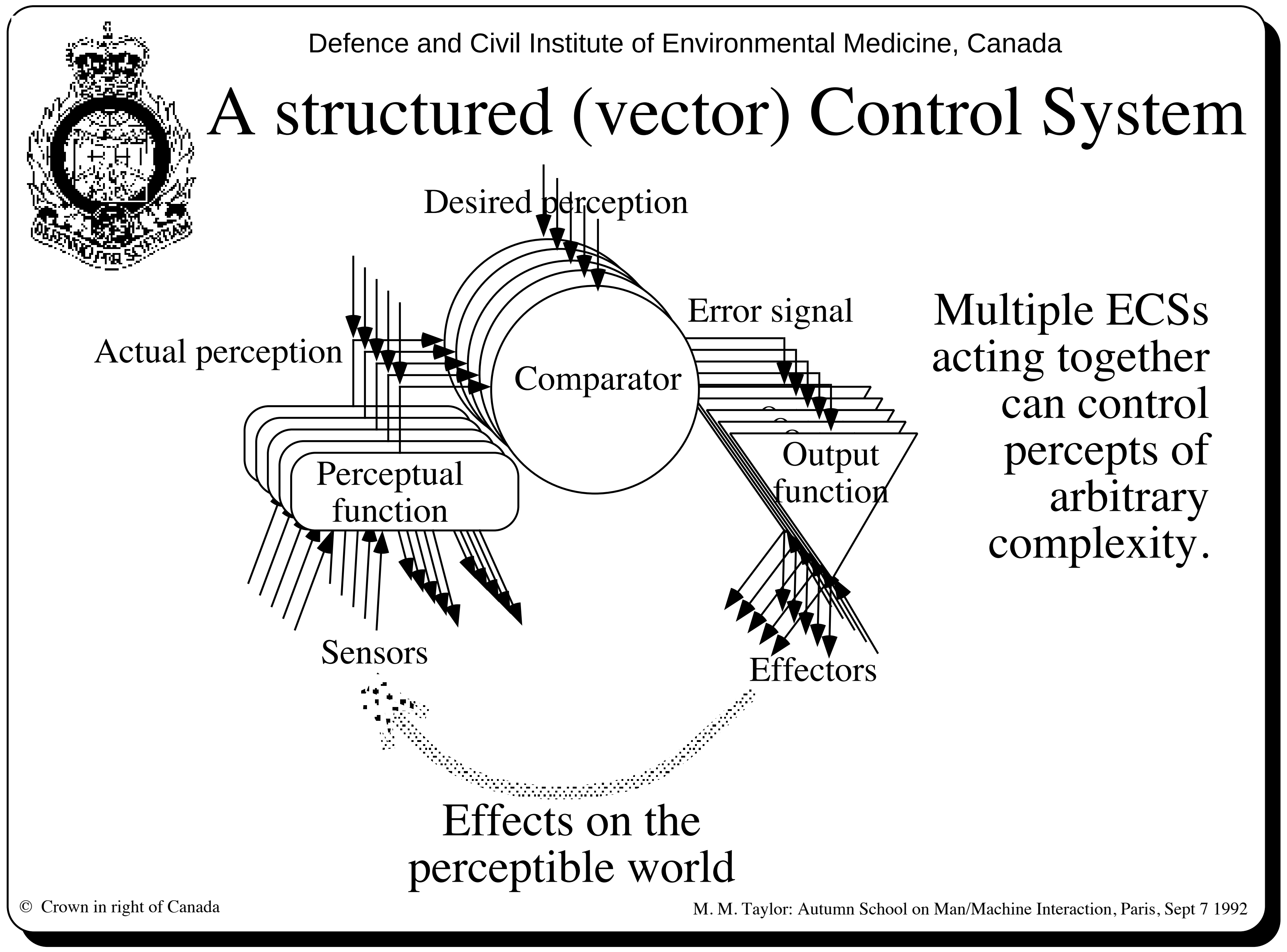
Two kinds of similarity controlling gain functions.
Each has a range of error over which it produces zero output.

These three figures represent general statements about PCT and Layered Protocol theory. The statements about task dependency apply more to LP in which the levels are those of dialogue, rather than to PCT, in which all levels are encompassed in the task.

The second figure should be thought about for a while. One might think the aphorisms are necessarily in agreement, or from another point of view that they are necessarily opposed.

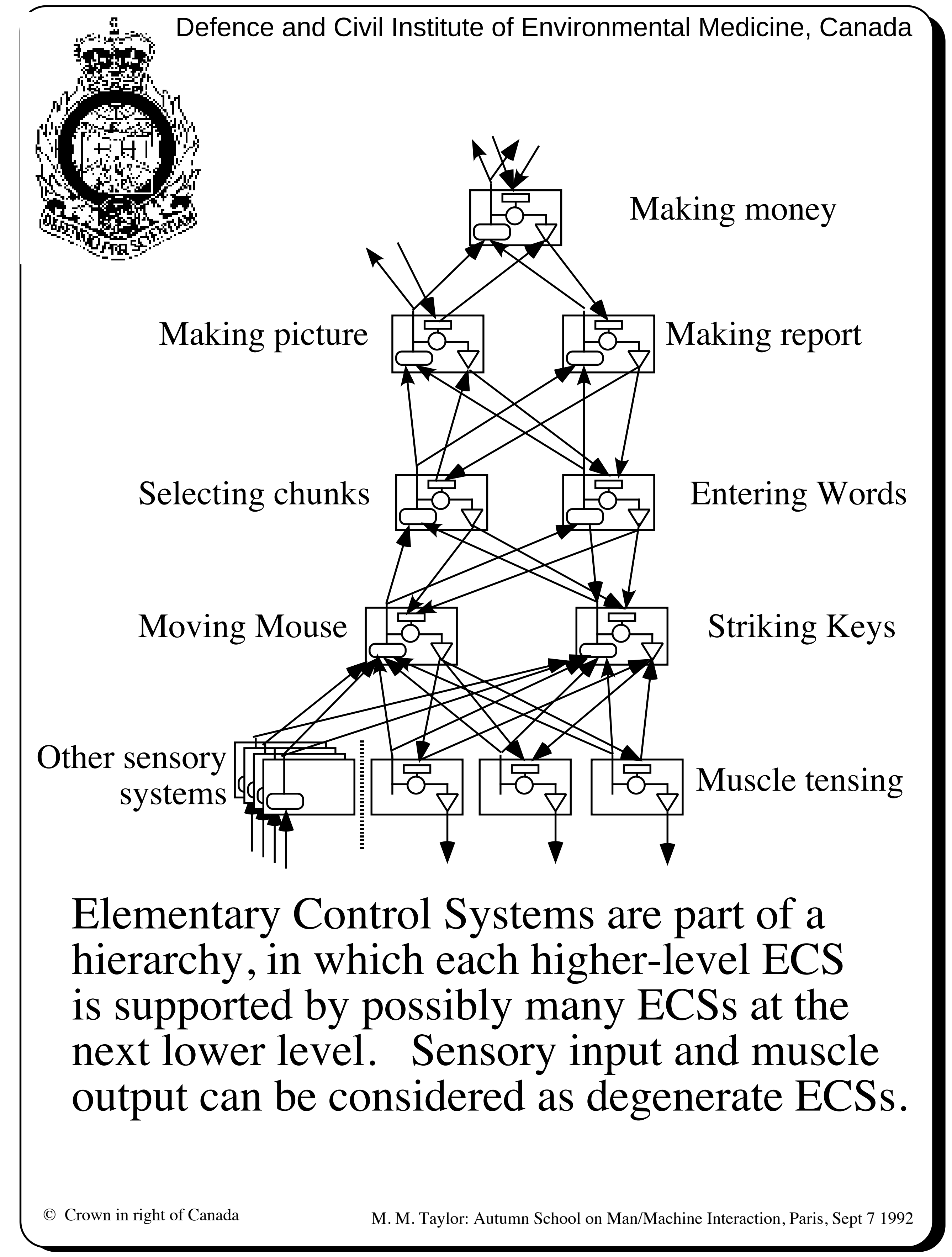
The different kinds of gain function represents the ways in which ECSs may react to error signals. In the standard version (top), any error is to be corrected, but in the other versions, a zero error is to be avoided (left—the get-away-from-danger operation), or a small error tolerated (right—the good is good enough; it ain't badly broke so I won't worry).

2-17



A collection of ECSs (Elementary Control Systems) that are working at the same time can be thought of as a Vector or Structured Control System (VCS or SCS). Together, they control a structured percept. Of course, none of the individual ECSs has any such construct (though the individual perceptual signals could in principle be brought together in the Perceptual Input Function of a higher ECS, to form a single perceptual signal representing the vector). It is convenient in discussion sometimes to treat a complex of ECSs like this as if it were a single control system in which each "signal" is a vector quantity rather than a scalar, and to draw the hierarchy exactly as if it were made of individual ECSs rather than SCSs. The distinction is ordinarily clear from context. One should remember, however, that even if we discuss structured control systems, the actual hierarchy is made from elementary control systems, the SCS being no more than a convenient way to talk about groups that we see as performing a coordinated function.

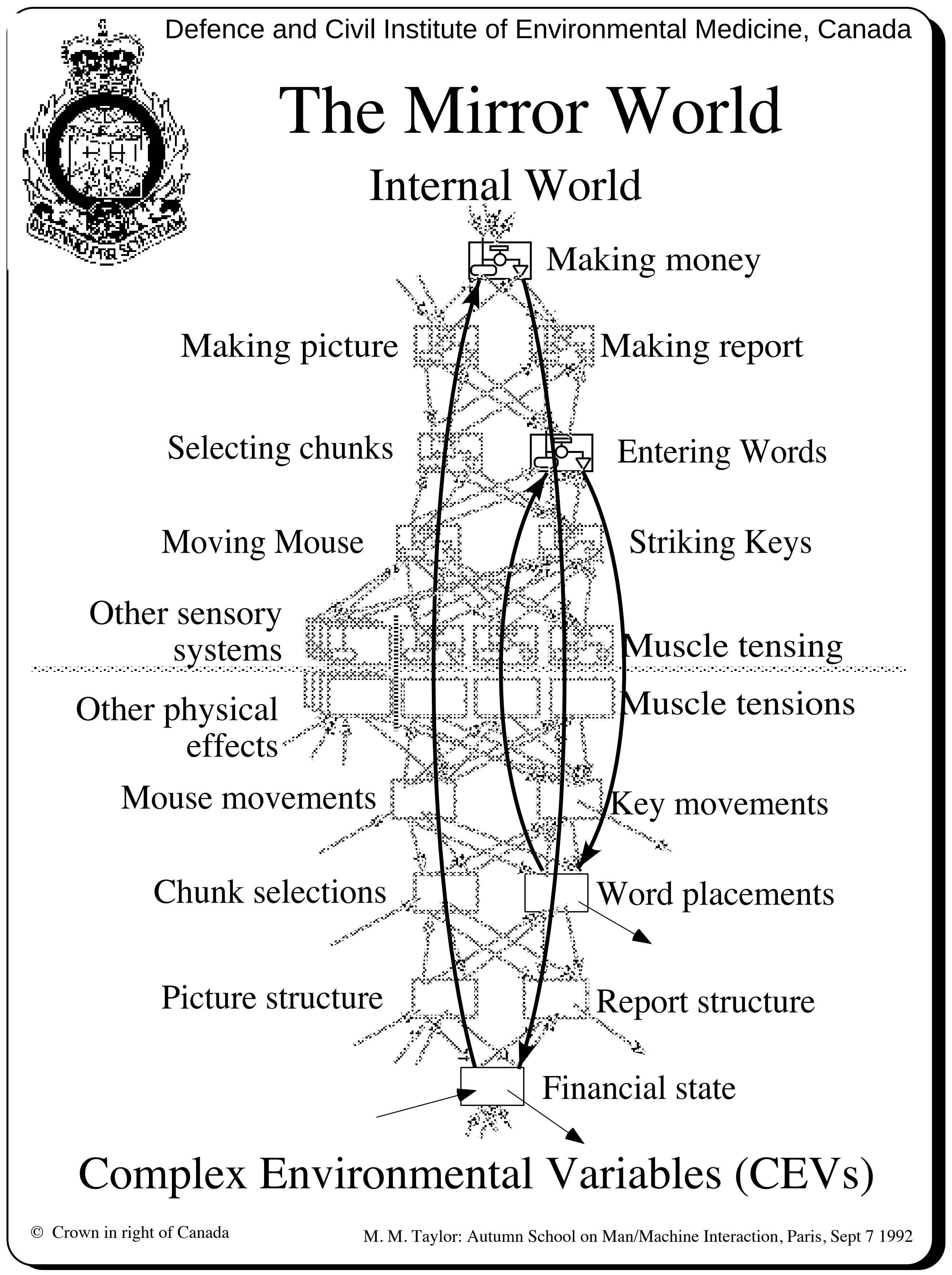
2-18



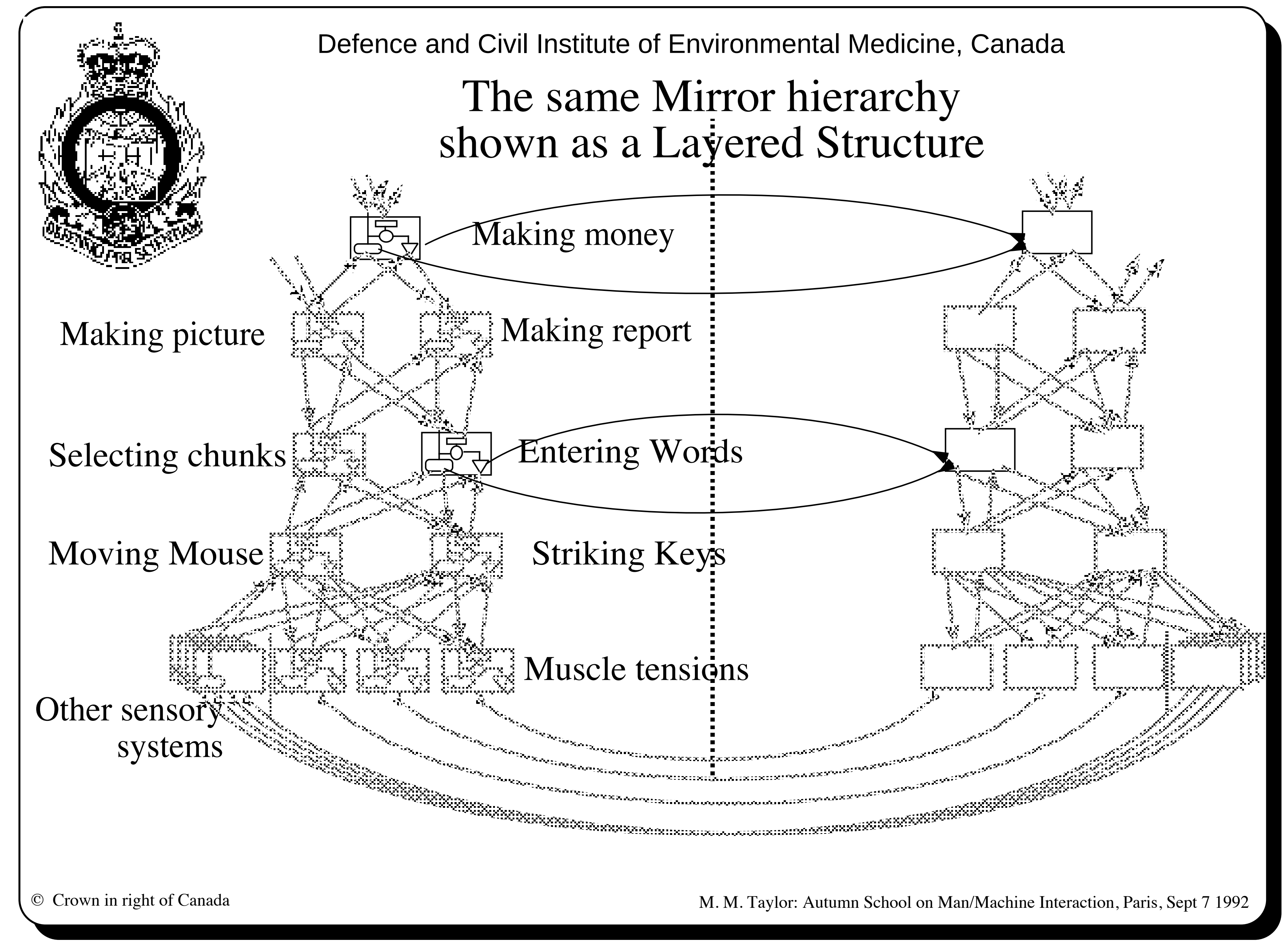
A sketch of a control hierarchy, with a few possible examples of perceptions that might be controlled as part of controlling for the perception of a good life (which in this person's view requires money, which is acquired through a job that involves making illustrated reports, which ...). The "ECS"s in this hierarchy are more probably SCSs, but since the structured perceptual signal in an SCS could be turned into a scalar perceptual signal in a next-level ECS, the distinction is unimportant.

The Mirror World: A Key pair of figures

2-19



2-20

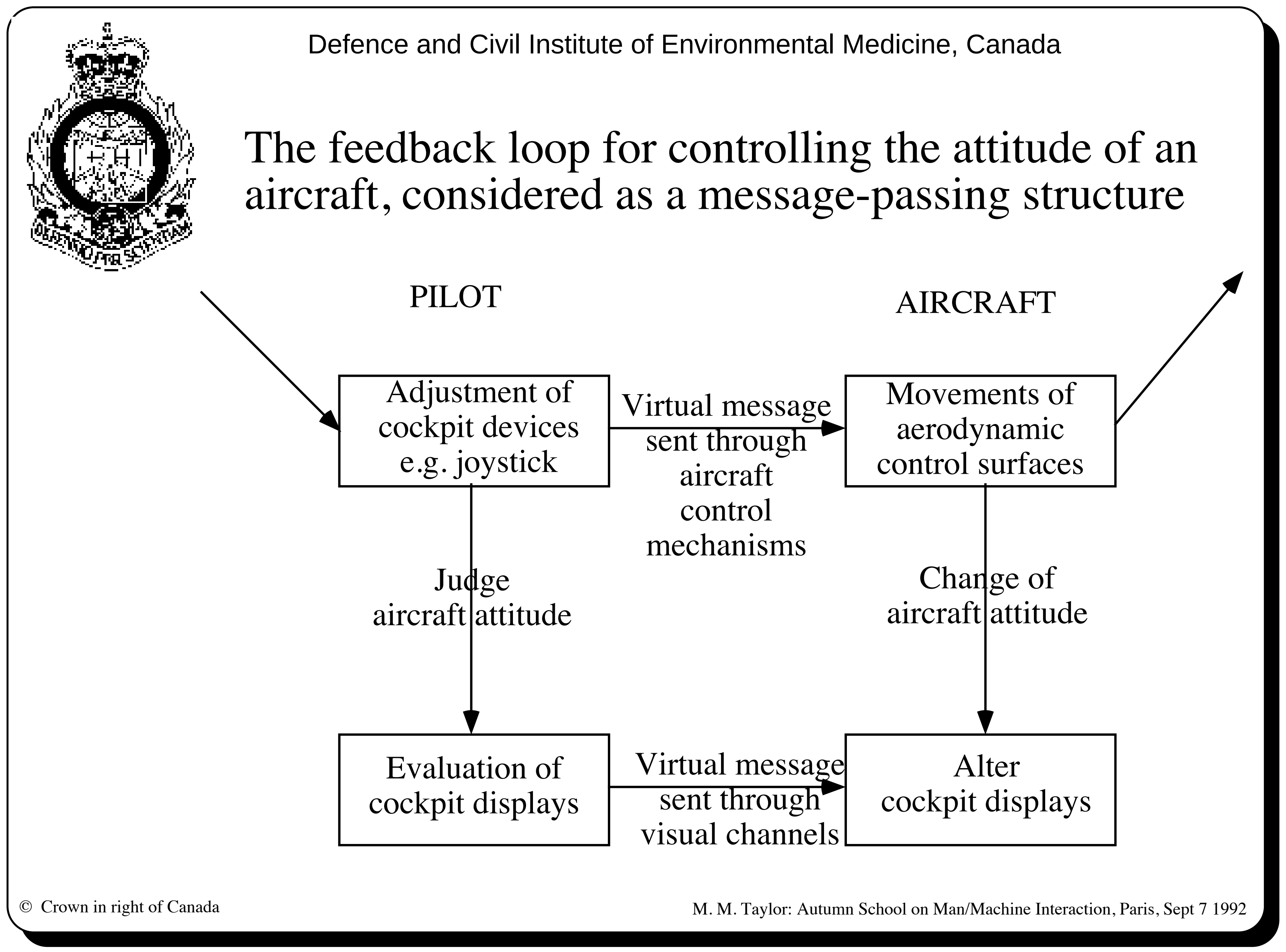


Each ECS controls some perceptual signal. But what in the world does this signal correspond to? It is the result of several levels of transformation through the various perceptual input functions, its own and those of lower ECSs. So, it is controlling some complex, probably highly non-linear function of its sensory input. It knows nothing about the world. It knows only the value of its perceptual signal, but that signal represents something in the world that can be affected by its output signals. The perceptual function can be related to something that we call a "Complex Environmental Variable" (CEV) in the world.

The second figure is identical to the first, except that the picture of the mirror world has been rotated to show more clearly the relation between an ECS and its CEV. On the output side, the outputs of the ECS are transformed into reference signals for lower-level ECSs that work on lower-level CEVs, and through those lower-level CEVs, eventually the one controlled by the ECS in question is affected. The sensory effects that allow the CEV to be perceived percolate from the CEV "through" lower-level CEVs controlled by lower-level ECSs to the sensor systems, and then back up through the perceptual input functions until the perceptual signal corresponding to the CEV is formed. ***This is the crucial relationship between the observer and the outer world***.

The perception of financial state is improved in part because the mouse moved appropriately in selecting a word used in creating the report.

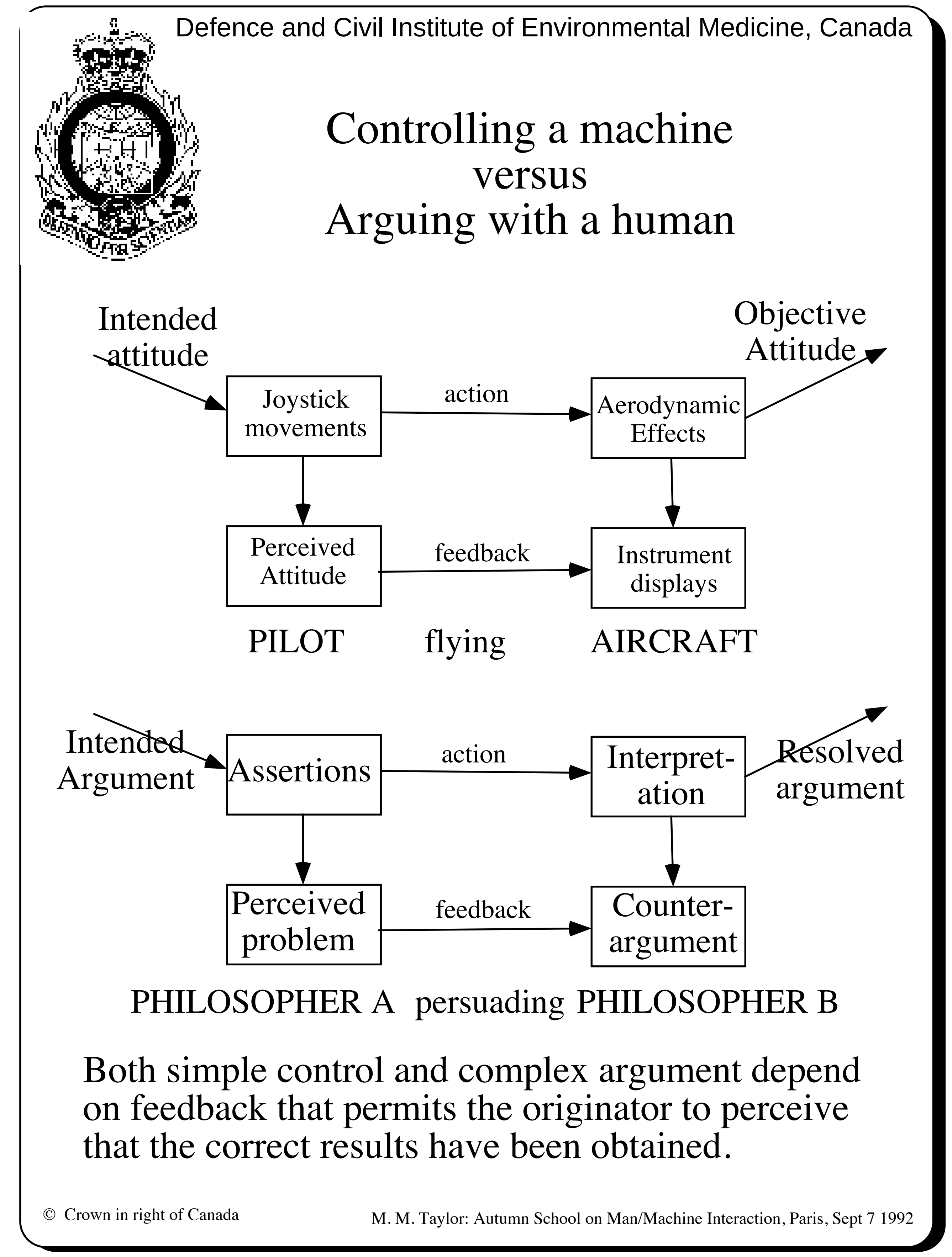
2-21



Consider a pilot controlling the attitude of an aircraft. We use a different visualization of the SCS (Structured Control System) and CEV, which will prove useful in our discussion of Layered Protocols. The arrow entering the upper left box is the reference signal for the SCS, or, in other terms, the desired state of the CEV.

The output signals from the SCS, though they actually percolate through many levels of ECSs before they affect the CEV, are shown as "virtual messages" that pass to the CEV. Their effects appear as "virtual messages" to the SCS's perceptual input function. The "virtual message" is a fundamental construct in Layered Protocol theory. It is convenient to distinguish the generator of the virtual message from the function that creates the perception of the import of the returned virtual message. Likewise it is convenient to separate out the corresponding (inverse) functions of the CEV, as if it were an independent control system controlling its perception, even though it may actually be simply responding mechanically.

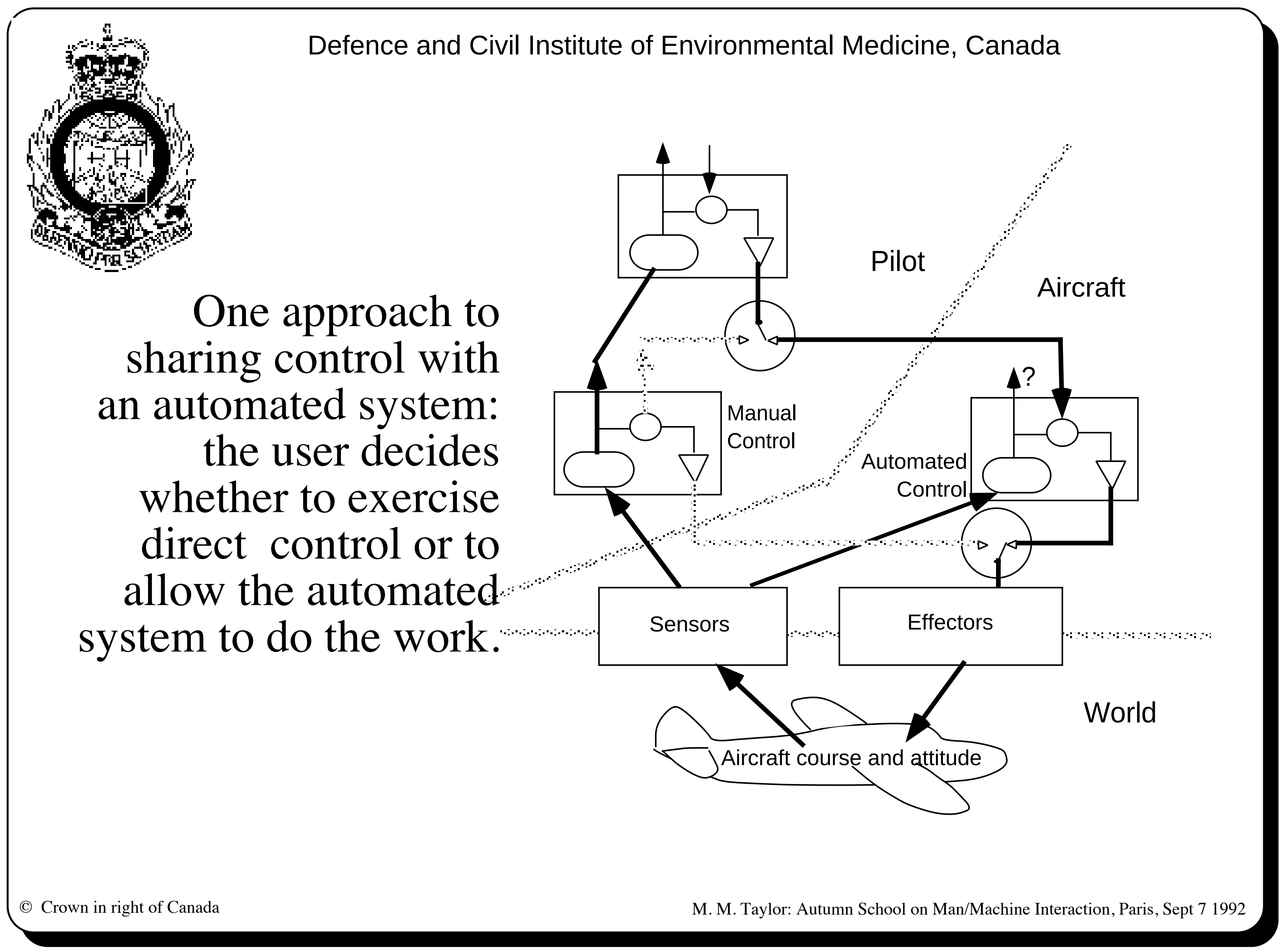
2-22

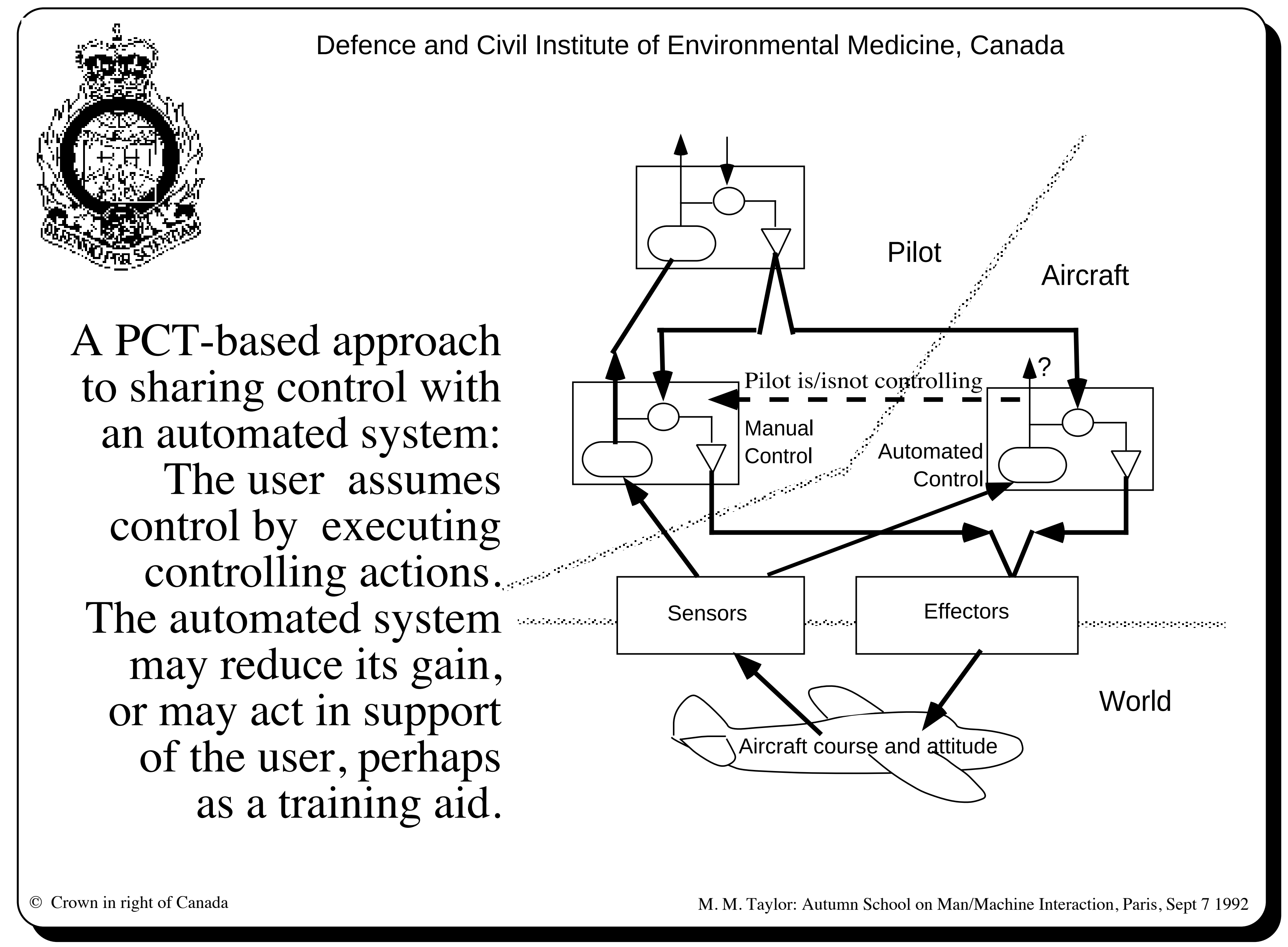


Control of the aircraft, looked at in this way, is very little different in principle from the feedback loop involved in getting across a point in an argument. In either case, the message originator continuously monitors the perception of the state of the CEV, whether that CEV be the attitude of an aircraft or the belief held by a conversational partner. This perception is compared to the reference perception, and differences lead to signals (messages) that affect the CEV.

In the next section of the talk, we discuss the Layered Protocol theory in its own terms, which these two figures should help to relate to PCT.

Supplementary Slides (not used in Paris talk)

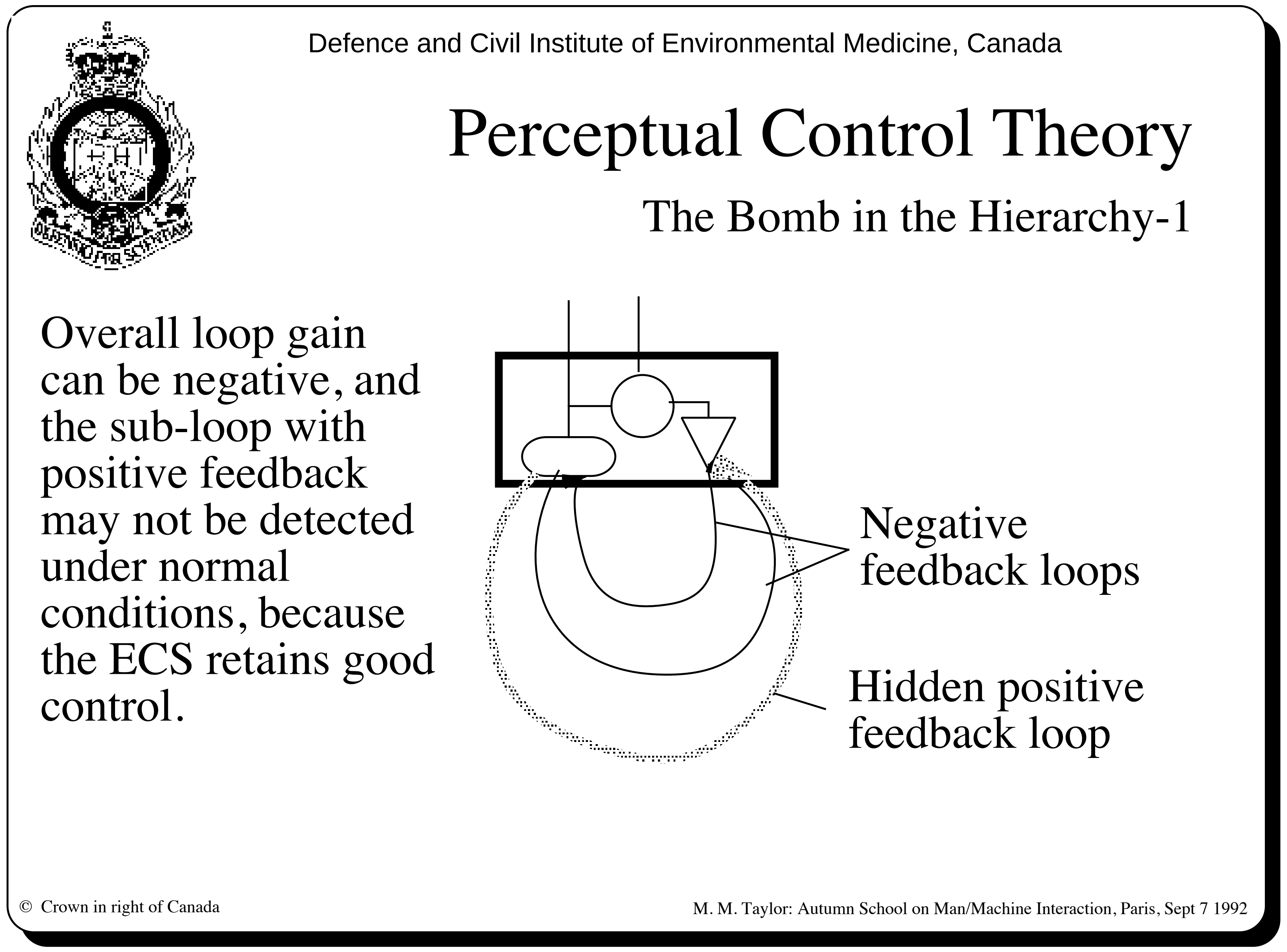


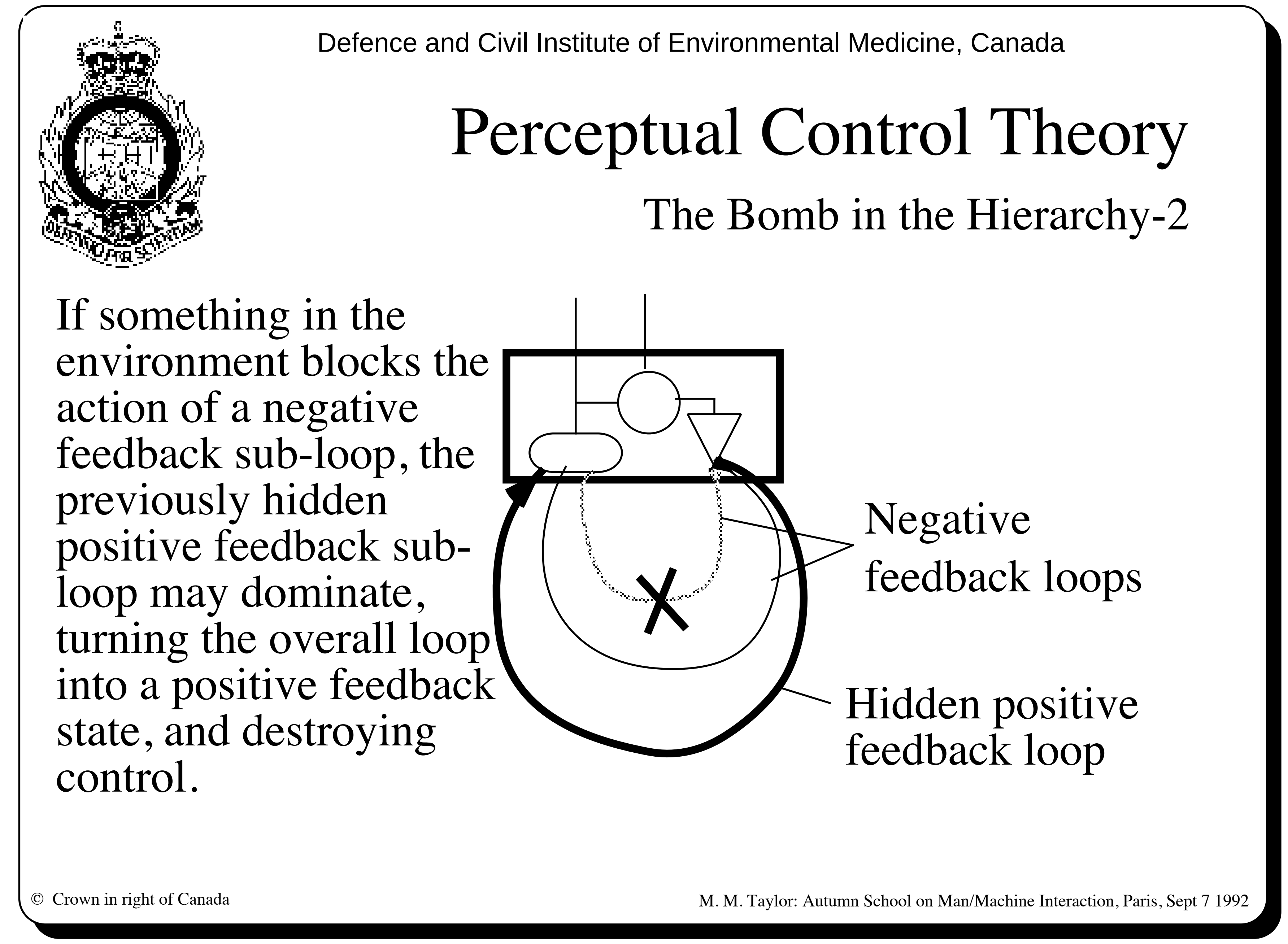


Graceful automation is a problem in aircraft control. Pilots like to have automated assistants, but nevertheless to maintain control of the aircraft themselves, particularly in case of emergency. A typical example of automation is the autopilot. The normal procedure is for the pilot to instruct the autopilot as to the desired heading and altitude, and then to relinquish control. There is a switch. Either the pilot or the autopilot is controlling the aircraft. In case the pilot wants to override the settings of the autopilot the switch must be reset (which can be done automatically). In PCT terms, the pilot sets the reference signal either for the autopilot or for his/her own muscular systems that deal with attitude control.

The PCT approach provides an alternative possibility that could be adapted to most forms of automation. If two ECSs affect the same CEV, it will ordinarily be more stable than if either ECS alone is controlling it. Each may cause the other to detect error, but the effects of external disturbance are resisted by both. Hence, the pilot and the autopilot could simultaneously control the attitude of the plane. The pilot would experience the plane as stiffer than normal, because of the autopilot. The effect of the autopilot is determined by its internal gain, which could be altered instantaneously as a consequence of sensing the pilot's actions, thus allowing the pilot to take over transparently, while retaining the stability of the plane when the pilot relinquishes control.

Supplementary Slides (not used in Paris talk)





The "Bomb" is an important aspect of a complex control hierarchy. Earlier, control hierarchies were discussed as if the sign of each link was adjusted so as to ensure that the feedback from output through the world to perception was always negative. This criterion can be met in a fully designed system working through a predictable world, but not in a system that develops through its varied interactions with the world. All that can be assured is that for an ECS functioning well under normal conditions, the overall feedback has come to be negative.

The overall feedback gain is based on the combination of many actions on aspects of the world that affect the sensory systems. It is quite possible for some of these actions, taken individually, to have undesirable positive feedback effects on the error. But any such positive feedback sub-loops are overwhelmed in an ECS that maintains good control by the negative feedback sub-loops.

Conditions in the world may change, blocking the effect of some of the desirable negative sub-loops. The overall feedback gain may then become positive, the previously hidden positive feedback sub-loop having been unmasked. The ECS causes actions that increase, rather than decrease its error. It "loses its temper because of frustration" due to the blockage of a normally available path to its goal. The path may be blocked because something fails that normally works, or because another independent control system is acting on an aspect of the world normally used in a negative feedback sub-loop, or for any of a number of reasons.

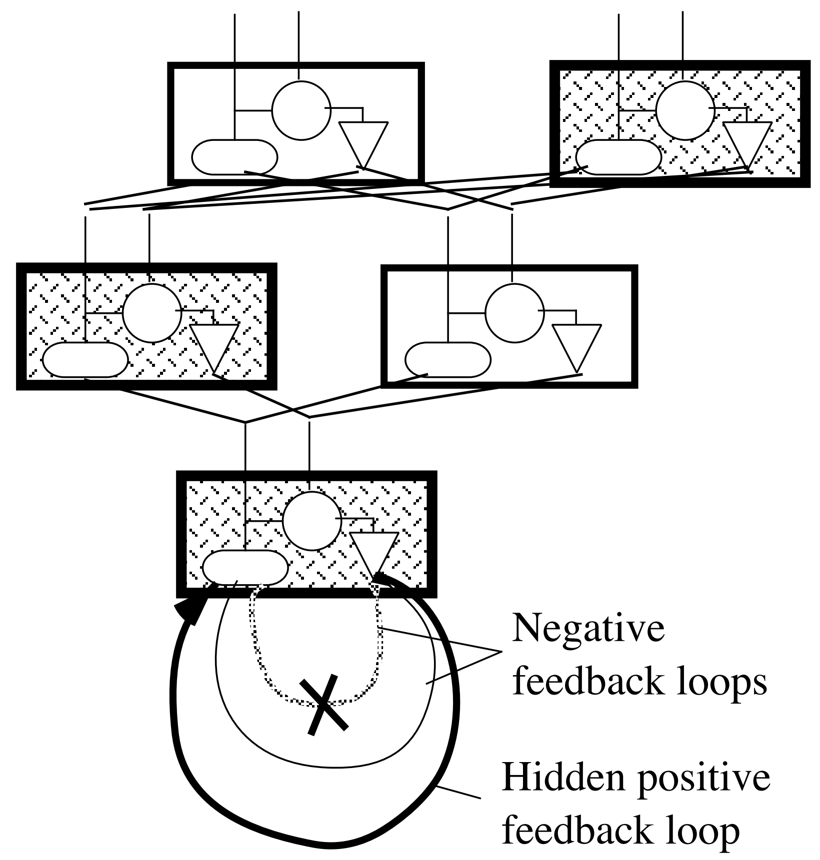
Supplementary Slides (not used in Paris talk)

Defence and Civil Institute of Environmental Medicine, Canada

Perceptual Control Theory

The Bomb in the Hierarchy-3

Positive feedback in one ECS could conceivably propagate up to higher-level ones that it supports, creating an avalanche of error in the hierarchy.



M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The Bomb can exist in any ECS, and is manifest through a path of action that involves lower-level ECSs. Some event in the world causes a hidden positive feedback sub-loop of some ECS to become manifest, and the overall feedback gain of that ECS becomes positive as seen from higher-level ECSs. Any ECS served by the "bombed" ECS then has a potential Bomb. If the other paths that serve it are not strongly enough negative, any of these higher ECSs may succumb, and go into a "bombed" positive feedback state. The Bomb can in this way propagate upward through the hierarchy like an avalanche, causing maladaptive behaviour at any level of abstraction.

Seriously maladaptive behaviour cannot last very long, or the organism will die. There must be a mechanism that reorganizes positive feedback loops that are revealed as the organism is exposed to different world circumstances. Over time, most bombs will be defused by this mechanism.

Defence and Civil Institute of Environmental Medicine, Canada

Perceptual Control Theory

The Bomb in the Hierarchy-4

Why would a "Bomb" exist? Inadequate learning of multiple means of accomplishing an end.

When would the "Bomb" show up? When some normally useful procedure fails for some reason.

What would be the symptom of "Bomb?" Frustration, leading to inappropriate and maladaptive action.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

A milder form of the Bomb can exist, in which a subloop does not contribute significantly to the ECS's perceptual signal. The output of the ECS, which overall moves the percept closer to its reference value, causes additional irrelevant side-effect actions—wasted effort or superstitious behaviour. These actions will be eliminated only if the wasted resources affect the ability of the hierarchy to control other perceptions. Many will be retained for the life of the organism.

Strong Bombs probably cannot last very long in a hierarchy that is exposed to a moderately disturbed world, but they can persist in a "coddled" hierarchy, one that is seldom stressed by exposure to unfamiliar or difficult circumstances. In such a "coddled" hierarchy, a Bomb is likely to be particularly dangerous and to cause a large avalanche when it explodes.

3-0

Part 3

Layered Protocols:

Protocol Loops and Protocol Nodes

3-1

Defence and Civil Institute of Environmental Medicine, Canada

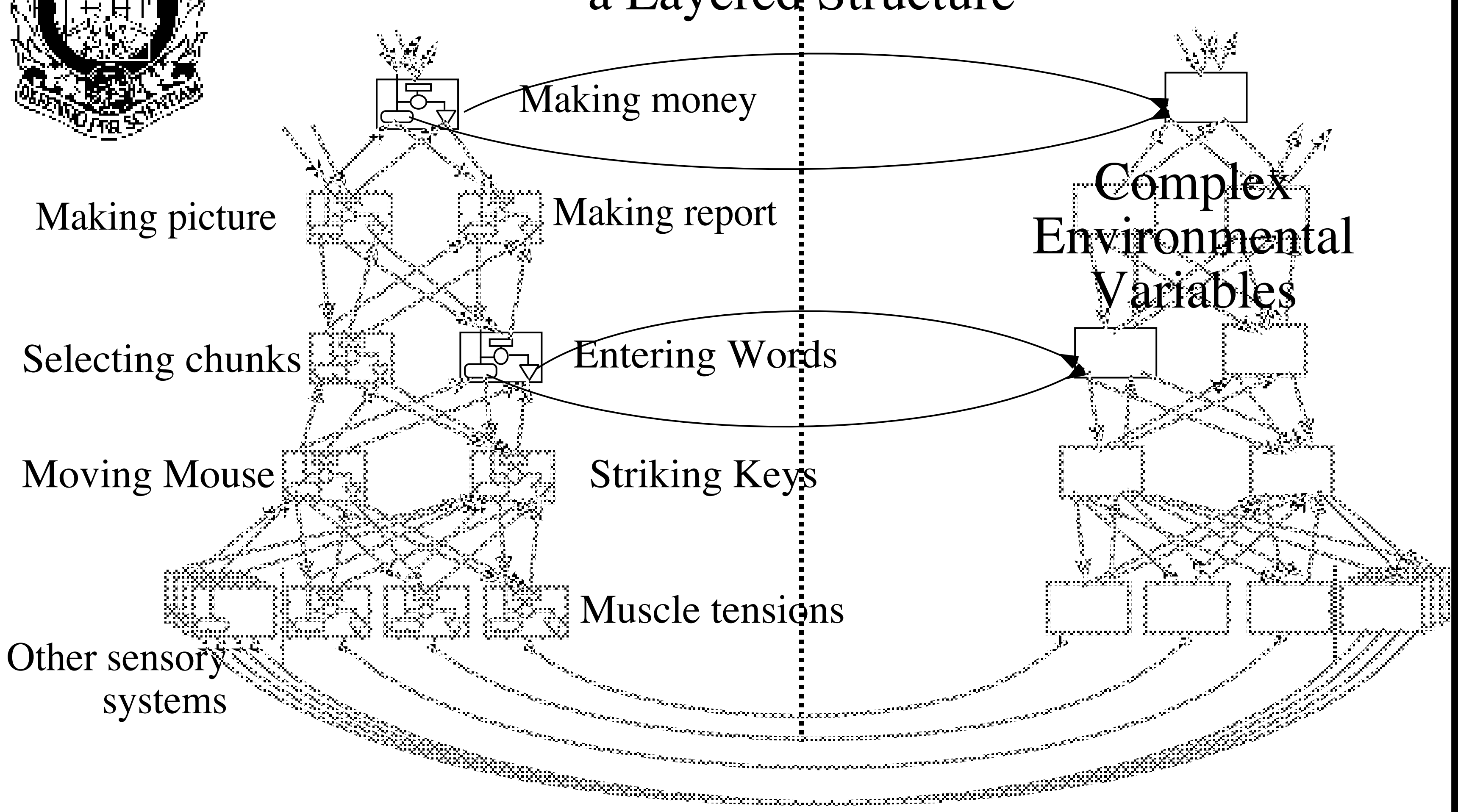
Basic Principle of Layered Protocols

All communication is the control of belief

3-2

Defence and Civil Institute of Environmental Medicine, Canada

The Mirror hierarchy shown as a Layered Structure



Part 3 of the talk introduces the basic ideas of Layered Protocol theory, and discusses a construct called a "Protocol Node" (PN). A PN takes the same role in LP theory that an ECS (or rather, an SCS) takes in Perceptual Control Theory. "Belief" in LP theory takes the place of "percept" in PCT, "communication" takes the place of "behaviour," and "message" takes the place of "effect."

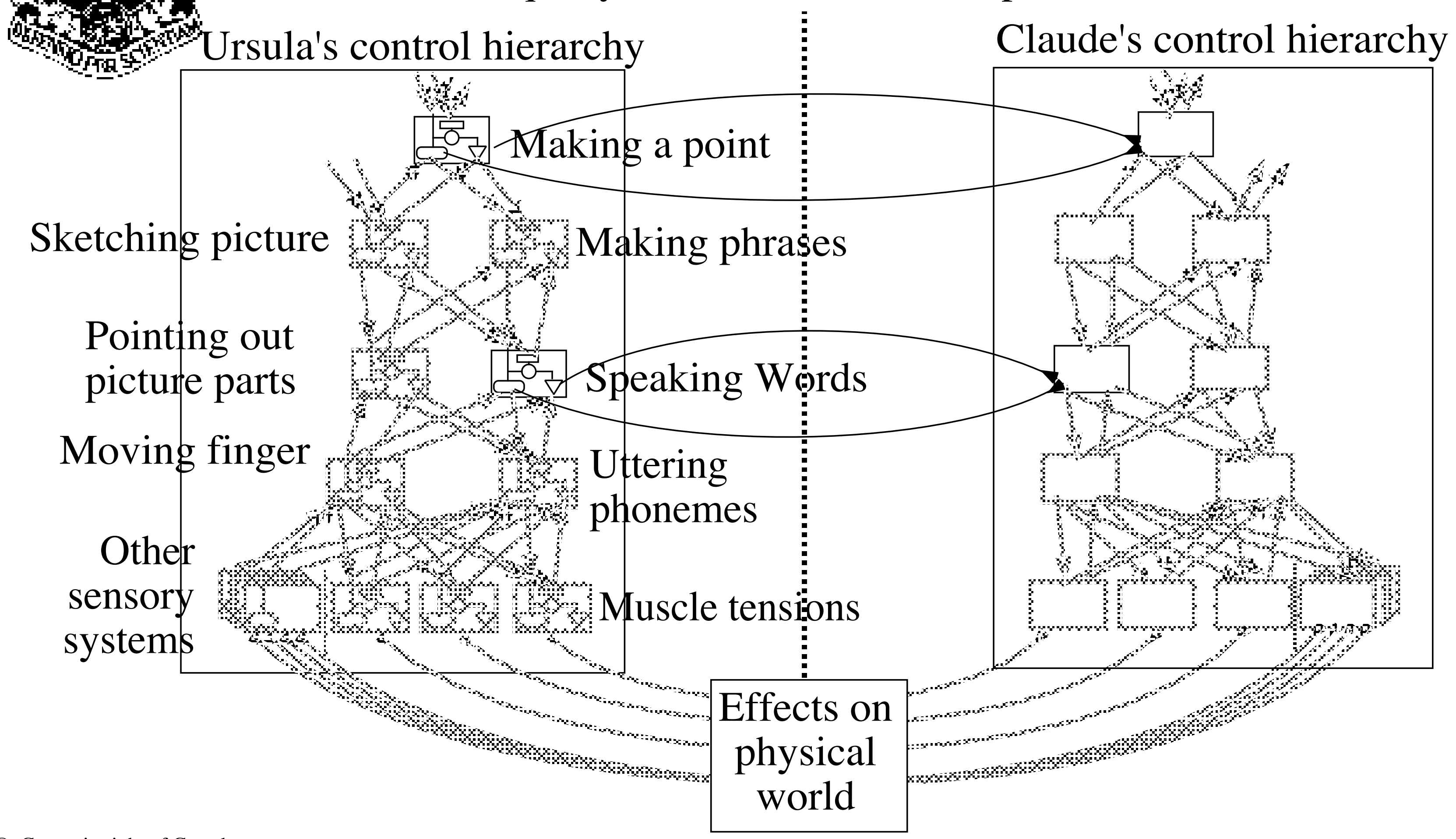
PCT shows how the percepts of various ECSs are linked through a mirrored hierarchy to complexes of elements of the real world that can affect the sensory systems (CEVs). Similar mirrorred hierarchies are at the heart of LP theory, except that in place of a CEV there is a Protocol Node (PN) in the communicative partner. The mirrored PN provides feedback consequent on the originator's messages in much the same way that a mirrored CEV in the world provides feedback consequent on the outputs of an ECS.

3-3

Defence and Civil Institute of Environmental Medicine, Canada

A Layered Protocol hierarchy between 2 people

(Ursula and Claude can be read equally well as User and Computer)



 M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

3-4

Defence and Civil Institute of Environmental Medicine, Canada

LAYERED PROTOCOLS

A principled approach to the design, analysis, and evaluation of interaction between humans and complex computational systems (or between humans).

Permits the integration of linguistic and non-linguistic channels of communication, and of disparate modes of communication.

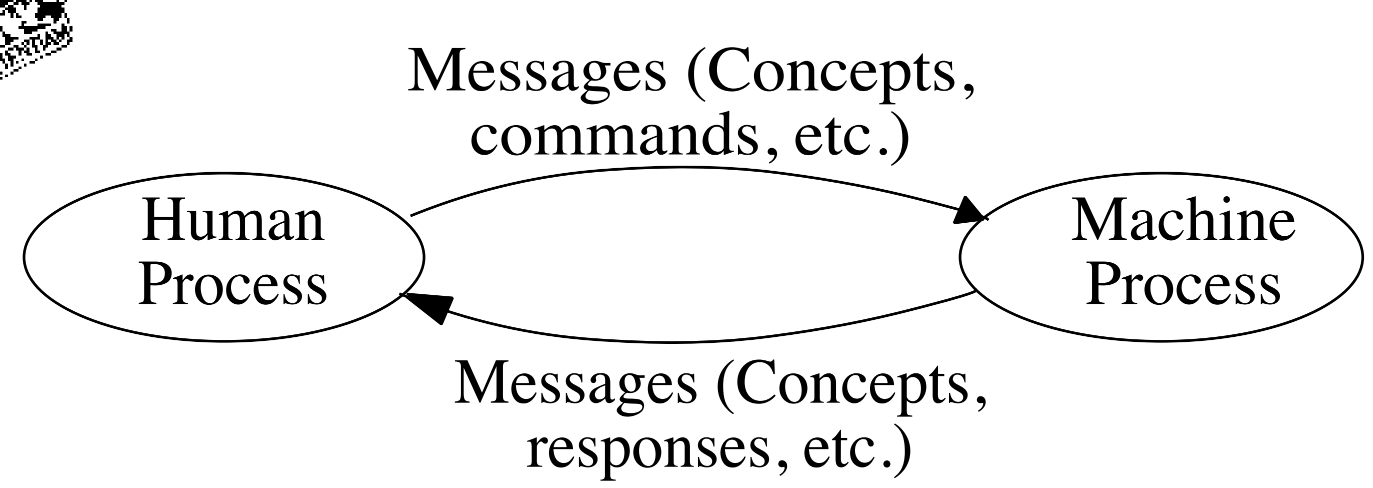
Incorporates the purpose of a communicative action as an integral element of the interaction.

 M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

3-5

Defence and Civil Institute of Environmental Medicine, Canada

The Telepathic Machine



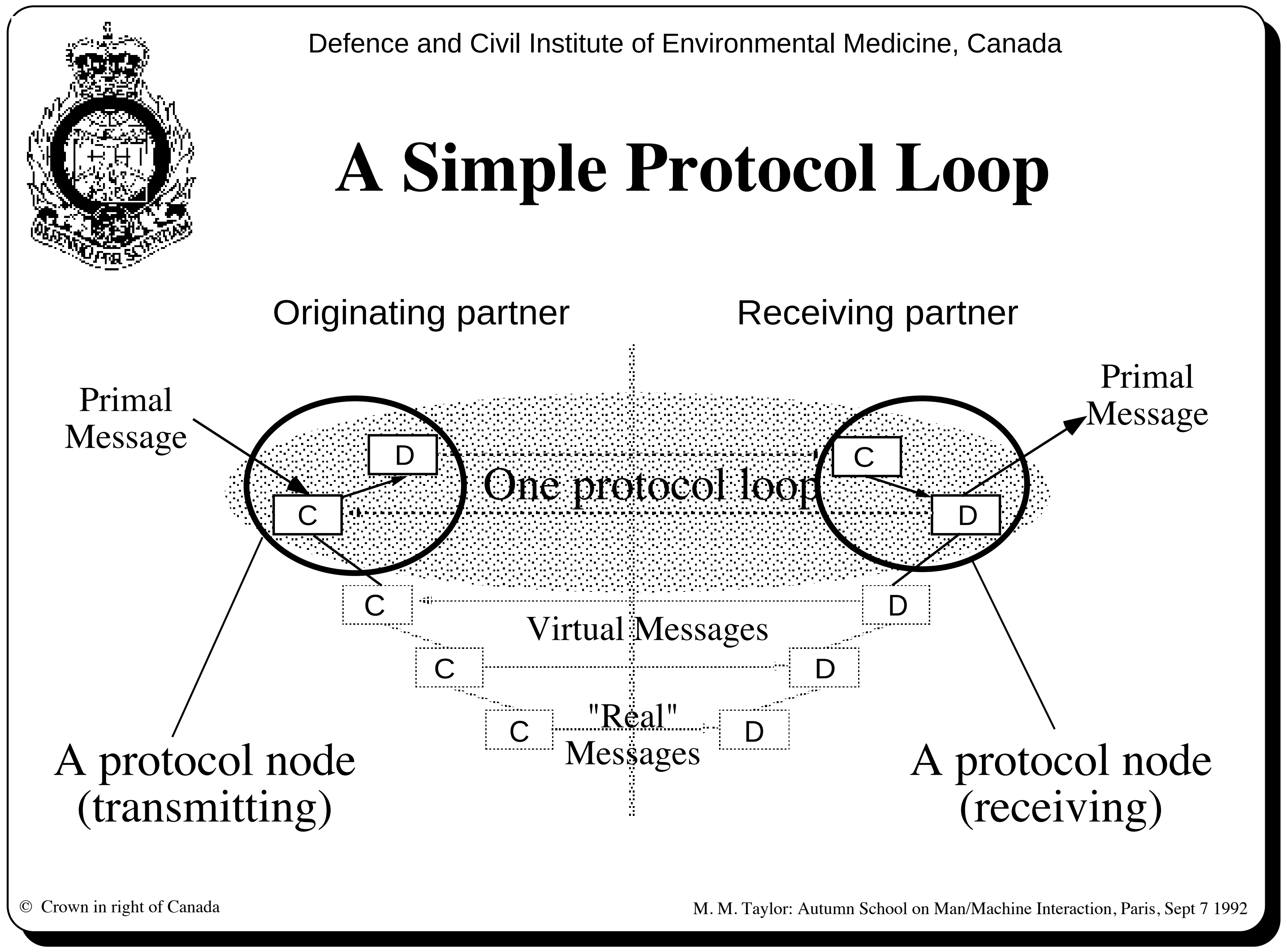
The ideal system, with mutual telepathy. The machine does what the human thinks of it doing, and the human appreciates directly what it has done. Real life is not like this. There must be an interface—controls and instruments, or computer I/O devices.

 M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Layered Protocol Theory is supposed to be a general theory of how and why people or quasi-intelligent constructed sytems communicate. It is a framework theory, into which details of the communication must be fitted in an interface design or in the analysis of different types of interaction.

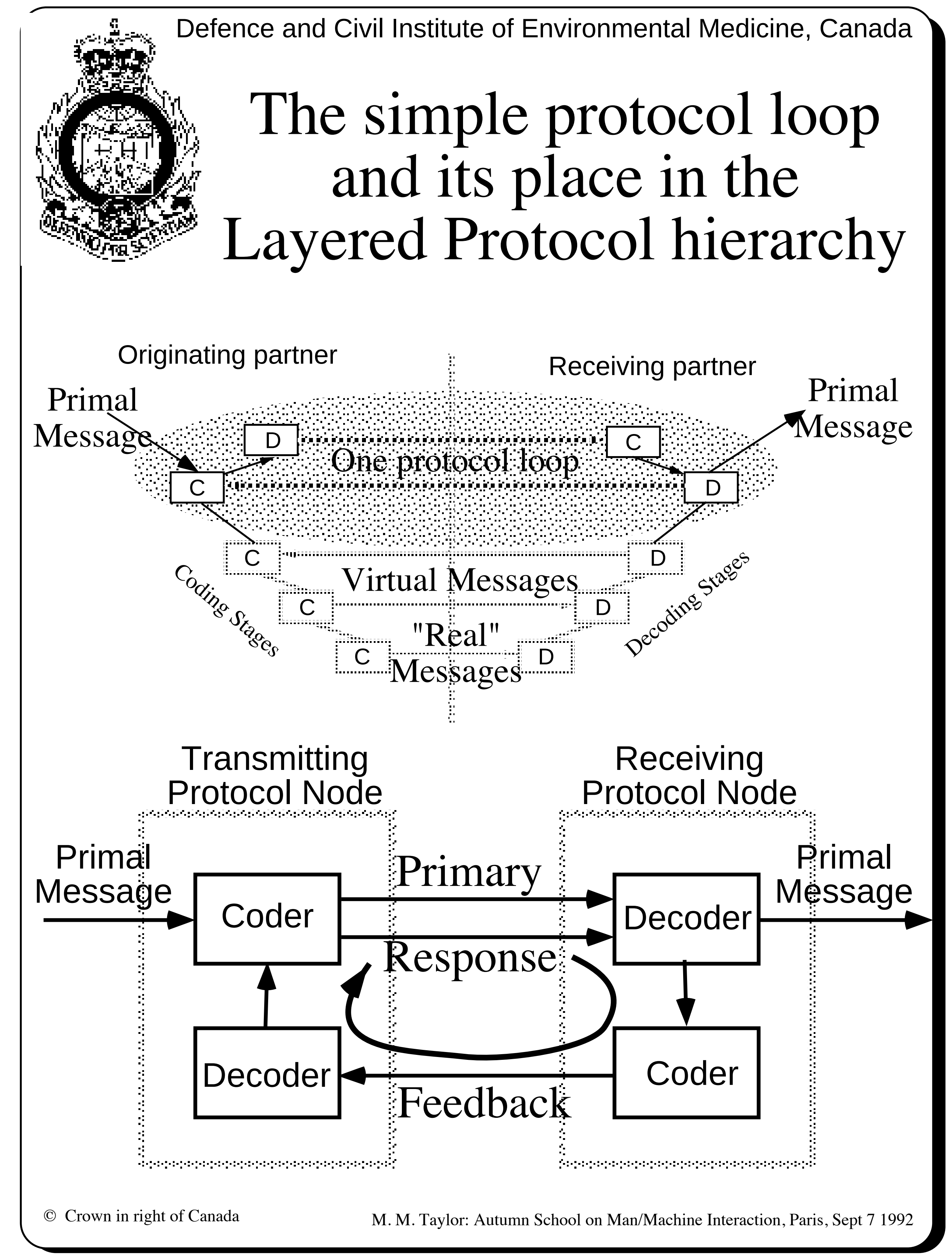
LP assumes that people want to communicate; that is, to bring a partner's state of belief to a desired condition. The partners are not telepathic, so they must use physical media to transmit the necessary information. LP does not consider (much) *what* they might want to communicate. It deals with how they do it, by the control of beliefs about the partner's beliefs at many levels of abstraction. This control is done through the communication of **virtual messages,** which include the necessary feedback.

3-7



The Primal message of a Protocol Node in LP theory corresponds to the Reference signal of an ECS in PC Theory. It is the information that the Originator wants the Recipient to have (i.e. the perception that the Originator wants to have of the Recipient's belief state). How the Originator gets the Recipient to have this information depends on what the Originator believes the Recipient already to believe. If the Originator believes the Recipient already to have the information, there is no need to transmit anything. Otherwise, something must be sent that will change into the desired state the Originator's perception of the Recipient's belief. That "something" is the virtual message that is the "encoded" version of the Primal Message. It is "encoded" according to some mutually acceptable Protocol. As part of the Protocol, the Recipient may use feedback messages to which the Originator may respond, until the partners agree that the message has been satisfactorily transmitted. The two-way circuit of messages is known as a Protocol Loop.

3-8



The part of the Protocol Loop that exists within either one of the partners is called a Protocol Node. It consists in its simplest form of a coder that determines the difference between the information the recipient should have and the information the recipient is believed to have, and a decoder that interprets the messages sent by the other party. The first message sent by the Originator's coder in this process is called the Primary message. The Recipient may send a feedback message to be interpreted by the Originator's decoder in light of the Primary message, and the Originator's coder may send a Response. The loop of feedback and response messages may continue indefinitely.

3-9

Defence and Civil Institute of Environmental Medicine, Canada

Protocol Nodes

A Protocol Node is the part of a protocol loop that exists within one of the partners

A Transmitting Protocol Node has a Coder that turns Primal messages into virtual message for transmission, and for constructing protocol (response) messages. It has a Decoder for interpreting protocol (feedback) messages.

A Receiving Protocol Node has a Decoder that interprets virtual messages, and for interpreting protocol (response) messages. It has a Coder for constructing protocol (feedback) messages.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The Protocol Node (PN) corresponds to the Elementary Control System in Perceptual Control Theory. In a cooperative dialogue, a PN in the partner originating the message works with one in the other partner. The originating partner's PN is called a Transmitting PN. It generates output that brings the originator's belief about the partner's belief closer to a reference state that we call the Primal Message. The other partner's PN is called a Receiving PN, for which the reference condition is the belief that the originator's belief is at its reference level, which is to say that the originator believes the virtual message has been satisfactorily transmitted. The Coder of either kind of PN outputs actions that are Primal messages for lower PNs, and these actions are interpreted as a virtual message by the Decoder in the partner's PN. Each Decoder continuously monitors the virtual messages from the other Coder, until it seems that the Primal Message has been successfully transmitted.

3-10

Defence and Civil Institute of Environmental Medicine, Canada

What is Dialogue in LP?

An exchange of messages within a Protocol Loop, which usually results in the transmission of a primal message, constitutes a dialogue.

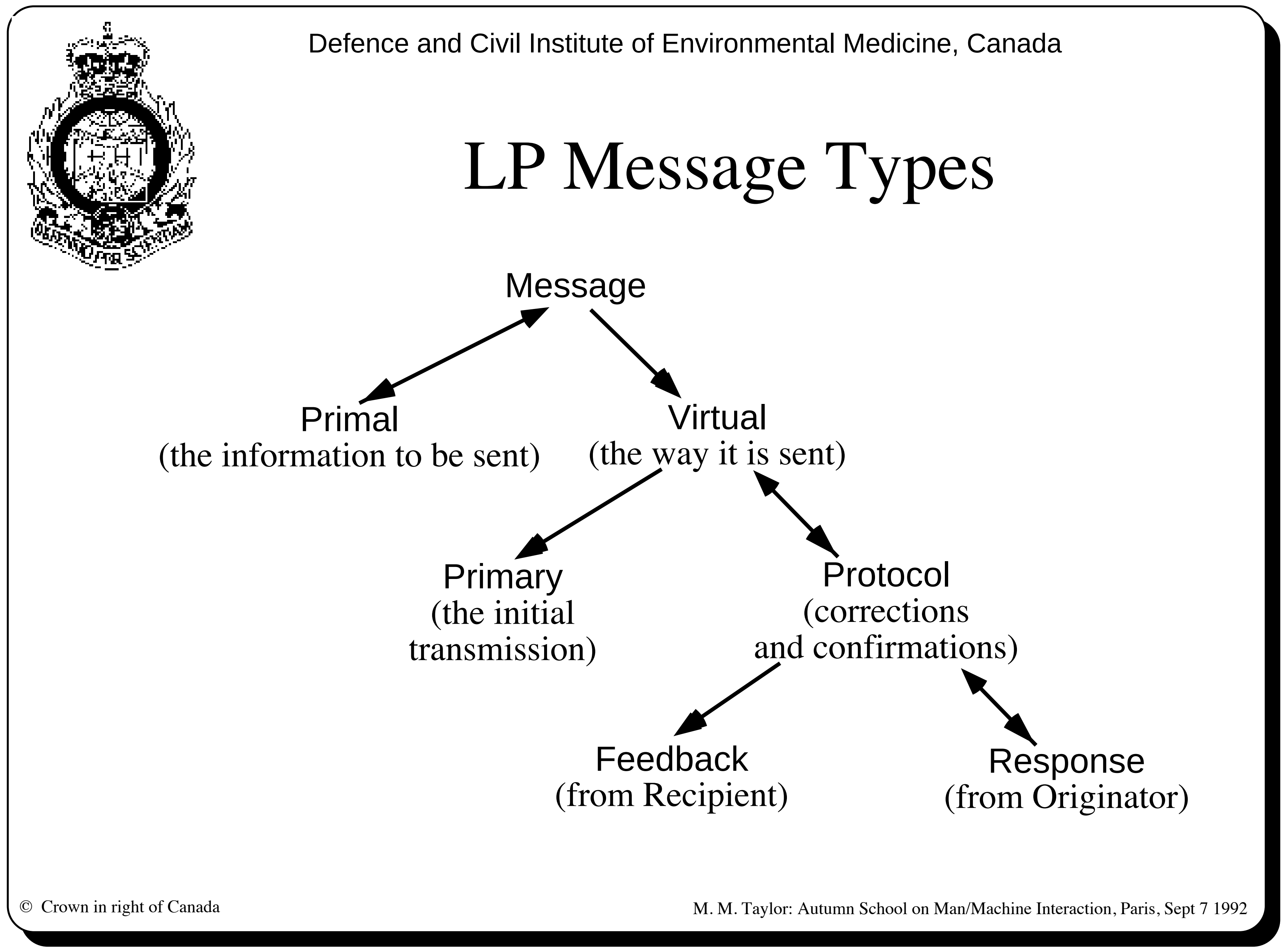
Each virtual message going one-way in a protocol loop forms one or more primal messages for a lower loop, and is therefore supported by a whole dialogue in the lower layer.

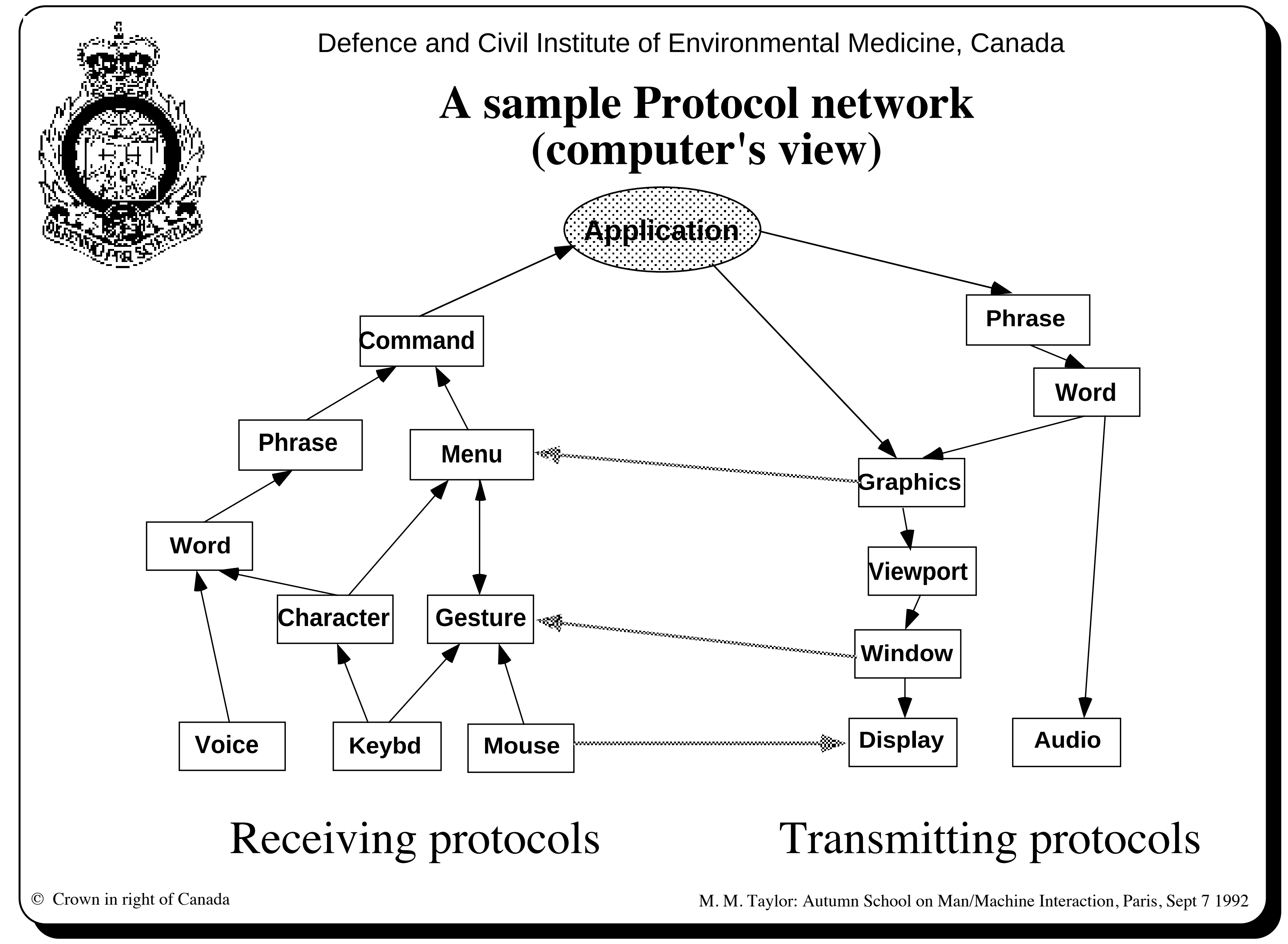
The total dialogue between the partners can be seen as the set of all the dialogues going on in all the protocol loops in all the layers.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The set of virtual message that go around a Protocol Loop to transmit a single Primal message is a dialogue. Messages in a dialogue do not need to follow rules of turn-taking. Indeed, it is normal for messages in the loop to pass in both directions simultaneously, so that the originating partner can continuously modify the actions required to bring the belief state to its desired reference value.

(Despite the simultaneous two-way communication normal in most protocols, dialogue is often analyzed as a sequence of verbal turns. This may be because the analyst records only the verbal component of the dialogue. In verbal dialogue, the feedback at the level of, say, propositions, is likely to be expressed in the form of head posture, nodding, or quasi-verbal "Mm," "Uh-huh," and the like, performed while the originator is talking. At higher protocol levels, the lack of turn taking in most dialogues is even more obvious.)





In talking about the messages in a protocol loop, we divide them into several classes. The classes of Virtual message differ only in the circumstances in which they are used, not in their nature.

A Primal Message does differ from a Virtual message. A Primal message of a PN is the intended perception the originator wants to have about some state or action of the recipient, whereas a Virtual Message is what is transmitted in order to achieve this end. All Virtual messages are Primal messages for lower-level PNs.

The Primary message is the initiating message of a protocol. At very low levels, it may be the only message, on the assumption that it will with high probability be correctly interpreted according to convention. At higher levels, most of the burden of conveying the Primal Message is carried by the subsequent "Protocol messages."

We sometimes distinguish Protocol messages from the recipient back to the originator as "Feedback," and the consequent messages from the originator as "Response." Since the transmission in each direction in a fluid interaction is more or less continuous, it is not always possible to make the distinction among successive feedback messages or successive response messages.

The example network shows some PNs that might be involved in an interaction between a human and a computer, in which the computer uses voice, keyboard and mouse for input, and audio and a screen display for output. The network is shown from the computer's viewpoint. The human has an equivalent network, in which the sense of the arrows is inverted and "transmitting" and "receiving" are interchanged.

4-0

Defence and Civil Institute of Environmental Medicine, Canada

Part 4
(very short)

LOCATE

A sample Application Domain

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

4-1

Defence and Civil Institute of Environmental Medicine, Canada

LOCATE objectives

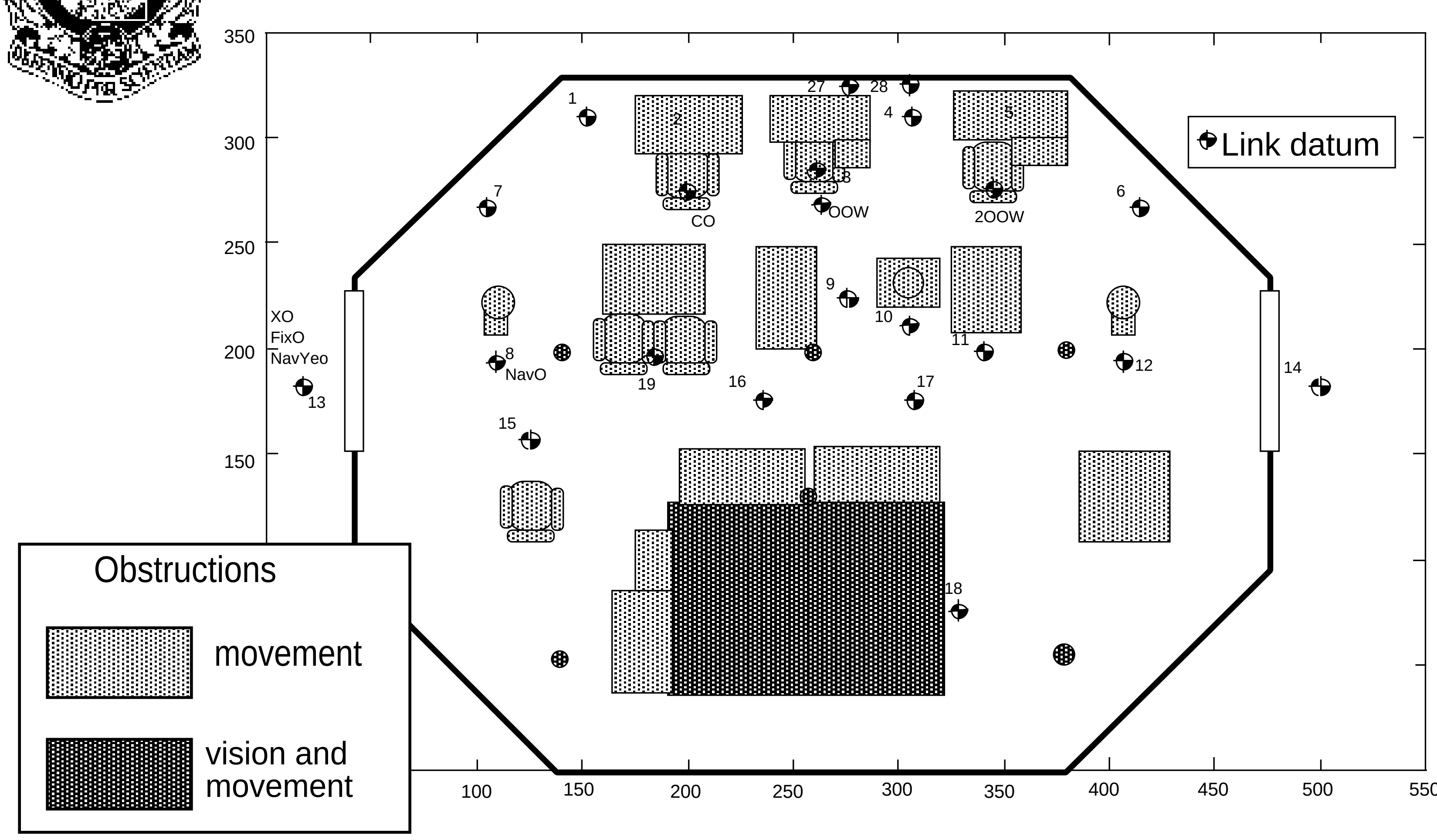
- LOCATE is the basis for a computer-aided workspace layout design tool
- Scale of problem is characterised as '...within the intermediate to far range of human sensory performance'
- Maximise communication in the visual, auditory and spatial (reach and movement) domains

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

4-2

Defence and Civil Institute of Environmental Medicine, Canada

DDH280 Bridge Study



M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

LOCATE is an existing program for evaluating the "goodness" of a workspace layout. It uses complex functions of several variables to create and evaluate links from one workstation to another, taking account of obstructions and the like in the environment. Such an evaluation is hard for a human, but optimizing one is hard for a computer, because there are many local optima in the very complex function that is derived from the many links. The existing program specifies the characteristics of the workspace in thousands of lines of alphanumeric data. We have chosen it as an example of an application that would benefit from an intelligent interface between the evaluation processor and the human designer.

LOCATE is discussed more in a later section. It is mentioned here only so as to provide a specific environment in which to think of some of the ideas introduced in the first half of the talk.

Defence and Civil Institute of Environmental Medicine, Canada

First Half Summary

All behaviour is the control of perception

The control of perception involves a hierarchy of elementary control systems (ECSs) that is mirrored in perceived structures and processes in the world called Complex Environmental Variables (CEVs).

All communication is the control of belief

The control of belief involves a hierarchy of Protocol Nodes that are mirrored in the communicating partner. Virtual messages pass between them in feedback loops to permit the transmission of information.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

These are the concepts I would like you to remember from the first half of the talk.

In the second half, I will go into more examples, and into more technical detail about different aspects of Layered Protocols.

II-0

Defence and Civil Institute of Environmental Medicine, Canada

Principles for Intelligent Human-Computer Interaction

Part II-Some Technical Aspects

Martin Taylor
Defence and Civil Institute of Environmental Medicine
Box 2000, North York, Ontario, Canada M3M 3B9
mmt@ben.dciem.dnd.ca

II-0

Defence and Civil Institute of Environmental Medicine, Canada

Part 5

Redundancy and Feedback

When is Syntax used?

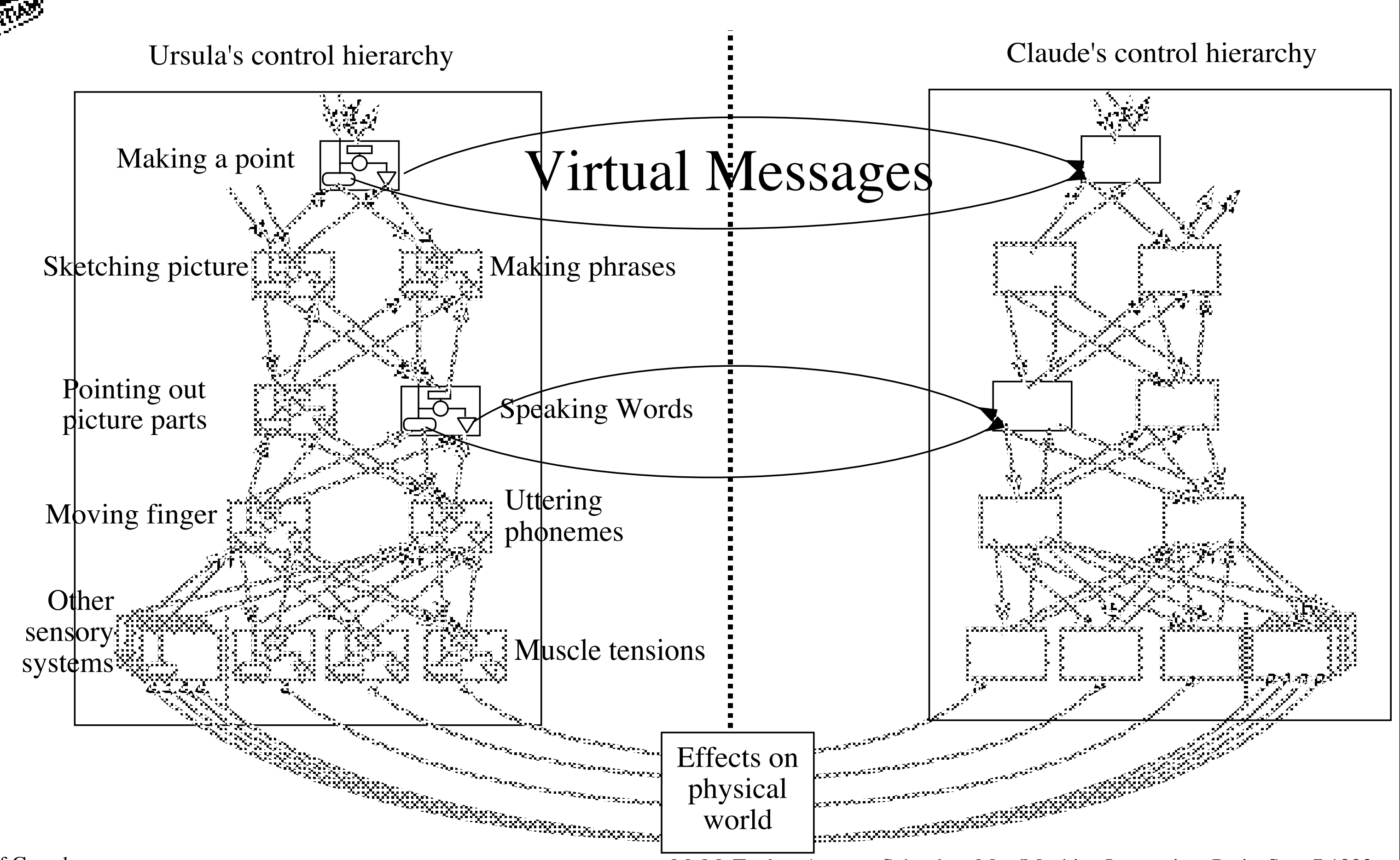
M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

5-1

Defence and Civil Institute of Environmental Medicine, Canada

A Layered Protocol hierarchy

(Ursula and Claude can be read equally well as User and Computer)

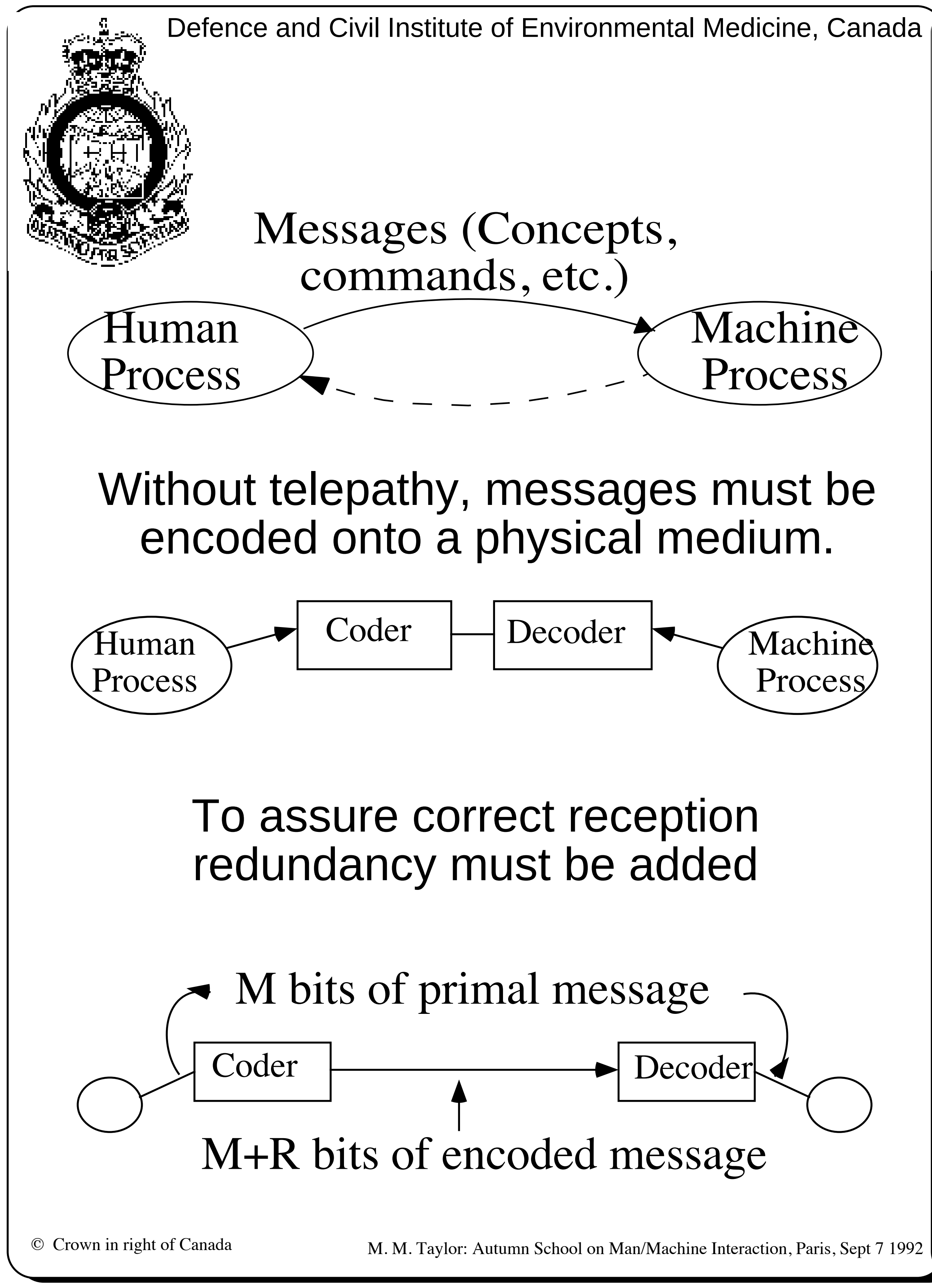


M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

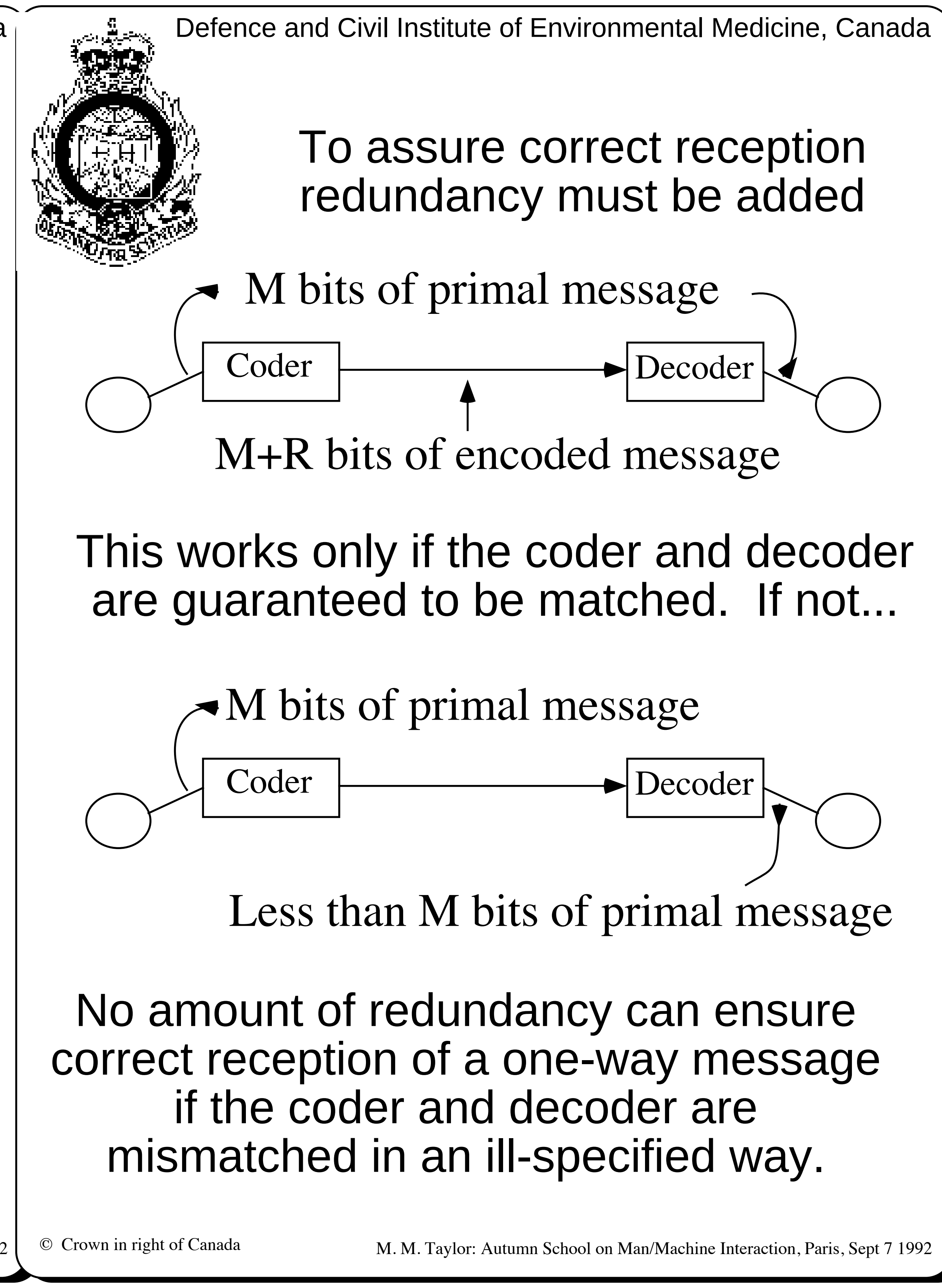
The second half of the talk consists of eight sections that sample different aspects of the Theory of Layered Protocols. Not much reference is made to Perceptual Control Theory, but it is always in the background if it is required to clarify some aspect of an issue.

The first aspect to consider is the role of redundancy in the Primary message, which is an aspect of what is known as *syntax* when the messages are considered as sequences of words. We take "syntax" to have an analogous meaning in protocols of any type, at any level of the hierarchy. When is it appropriate, and why? How do syntax and the use of feedback relate to one another at the different levels of abstraction represented by the layers of a hierarchy? It turns out to be all a matter of time constraints. Stable communication requires fast feedback and correct individual messages. Syntax helps correctness, but slows their passage.

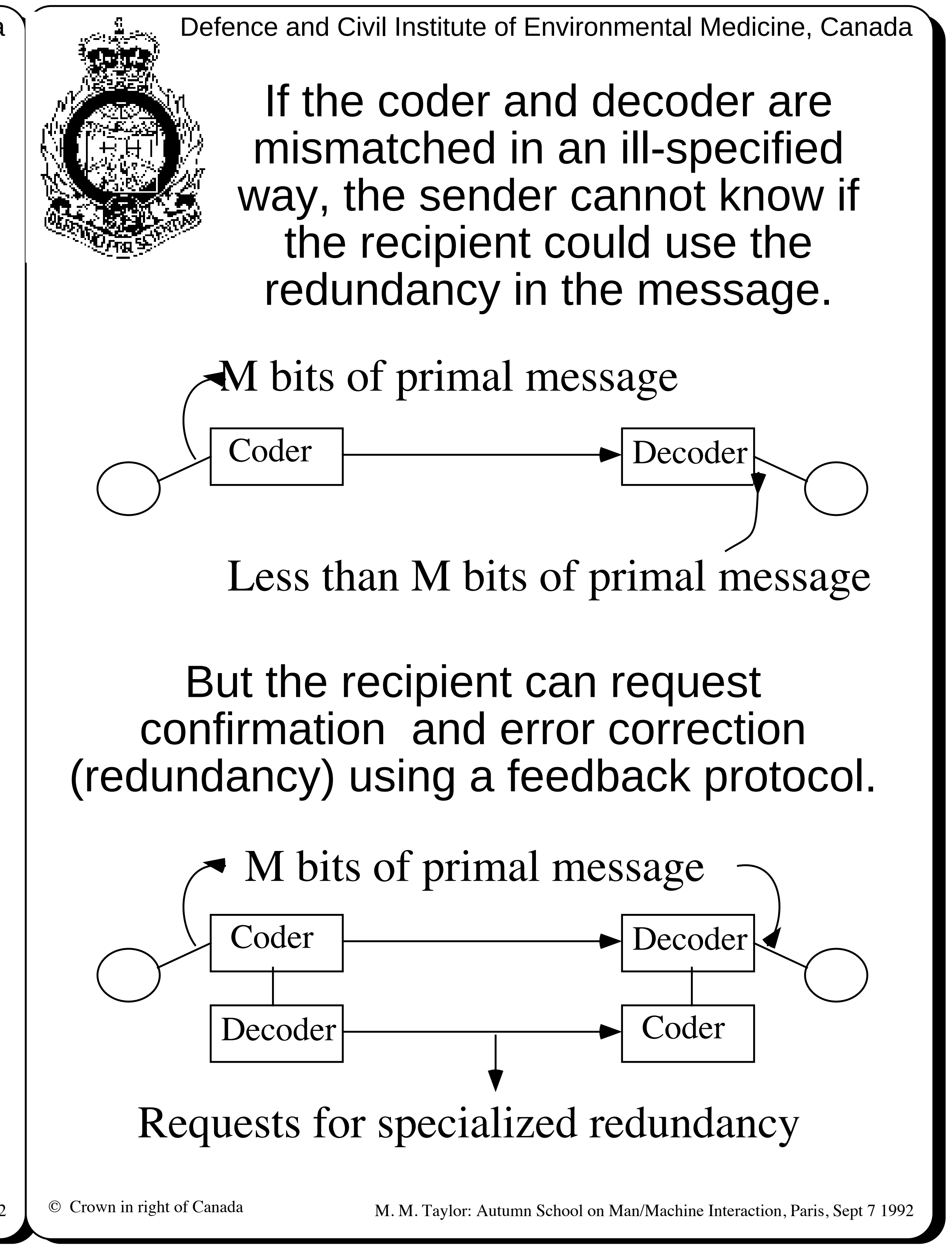
5-2



5-3



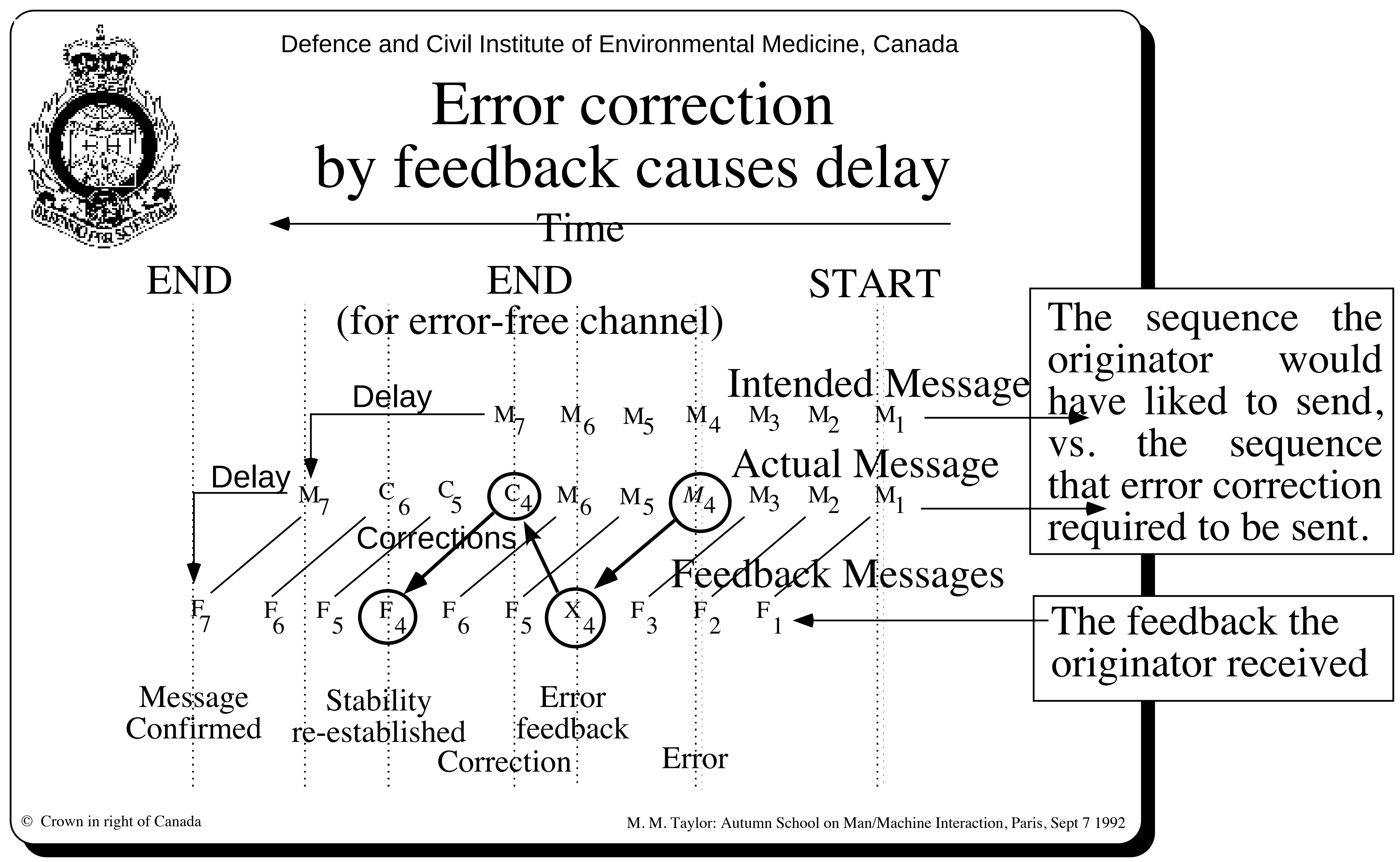
5-4



Redundancy refers to a situation in which not all of the conceivable patterns that could be transmitted have equal probability. If they do not, then the recipient requires less information than might have been transmitted to identify what was actually transmitted. If to identify the pattern would have required M bits of information, but the channel could have transmitted M+R bits, the amount of redundancy is R bits. The R bits of redundancy can be used to correct for transmission errors, but **only if the receiver knows the probabilities of the patterns that could have been sent**. In an extreme case, only certain patterns are legal, and the reception of an illegal pattern indicates that an error has occurred. Rules specifying the legality of patterns are often called the *syntax* of the pattern generator (Part 11 of the talk discusses another role of syntax).

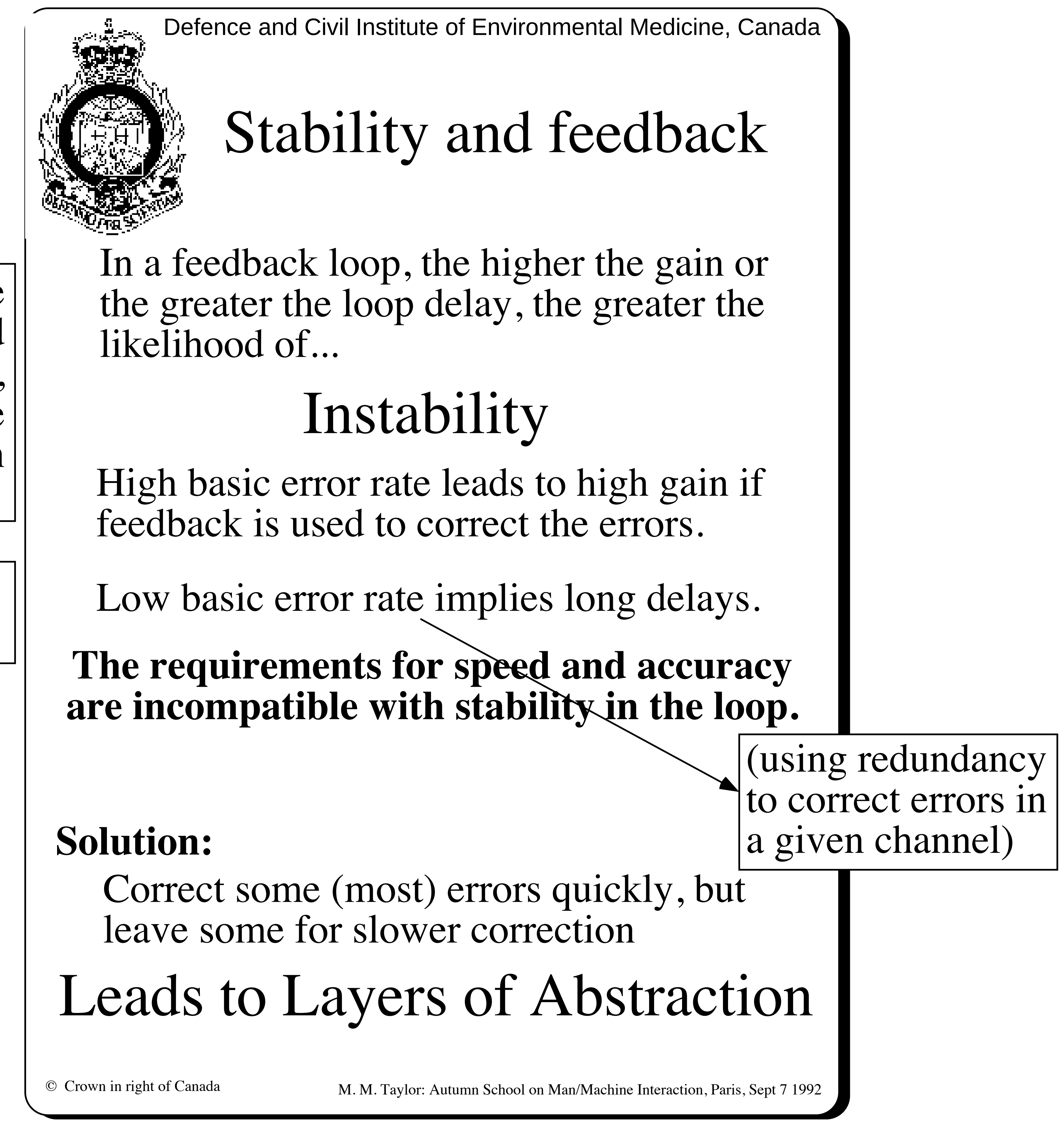
Syntax works if the two partners can agree as to the rules that the Originator uses to generate the transmitted message based on the intended content. There may be conventions that govern these rules, such as the grammar taught in schools. In extreme cases, such as programming languages, the conventions are precise, but more commonly they derive from common practice observed during long periods of interaction among partners of a common "kind," and thus cannot be guaranteed to be the same for the Originator and the Recipient. If they are not the same, then no amount of redundancy can ensure that the Recipient receives the content that the Originator intended to send. Simple "encoding" cannot work as a way of passing messages. Feedback, which can be considered as a request for specialized redundancy, is required.

5-5



Correcting an error in a message may cause delays longer than just the delay of the correction itself. This example shows a message composed of seven elements. The recipient sends a feedback message after each element is received, but there is some delay in both directions. In this example, the fourth message element causes the recipient a problem, but by the time the feedback message is interpreted by the originator, the sixth element has already been sent. The originator follows it with a corrected version of the fourth, but then must confirm or correct the fifth and sixth. Only after they have been checked can the final element be sent. Frequently, more than one feedback and response message is needed to sort out an error to the satisfaction of both partners. The number of such protocol messages required is analogous to the gain of a feedback loop, and affects its stability. The more feedback and response messages are required for each primary message, the less the output depends on the input of the loop, and the more it depends on the characteristics of the protocol itself.

5-6

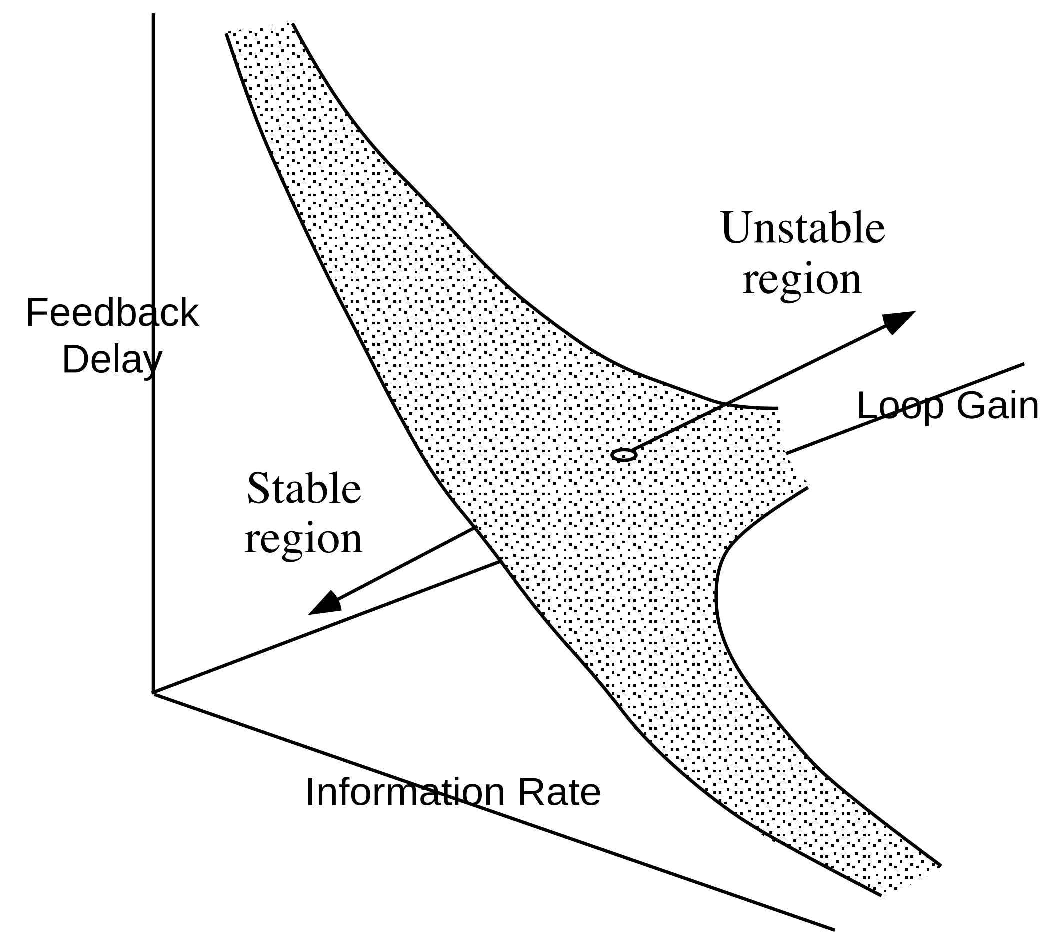


In a real world with ongoing events relevant to the dialogue, it is important that messages be sent and interpreted quickly. Furthermore, in any feedback loop, long delays tend to induce instability, particularly if the gain is high. But rapid interpretation increases the likelihood of error, thus increasing the equivalent gain of the feedback loop. The solution evolved by nature is to correct or generate feedback quickly for those errors that can be corrected quickly, and to leave others for slower systems to handle. The next few slides indicate how this dilemma leads to the concept of a hierarchy of virtual messages at different levels of abstraction, and thus to the Layered Protocol structure.

5-7

Defence and Civil Institute of Environmental Medicine, Canada

Criteria for stability of a feedback loop

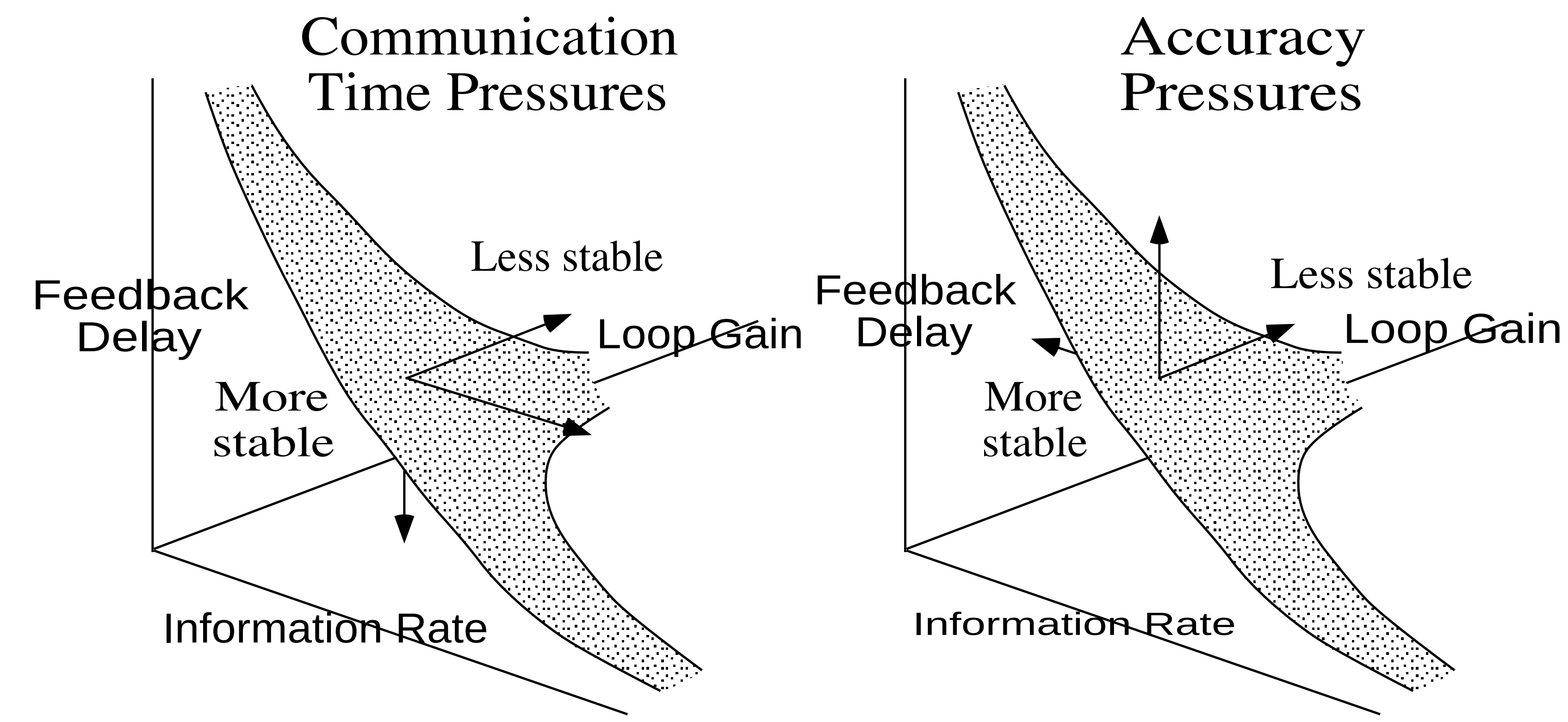


M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

5-8

Defence and Civil Institute of Environmental Medicine, Canada

Pressures opposing stability in a real interface



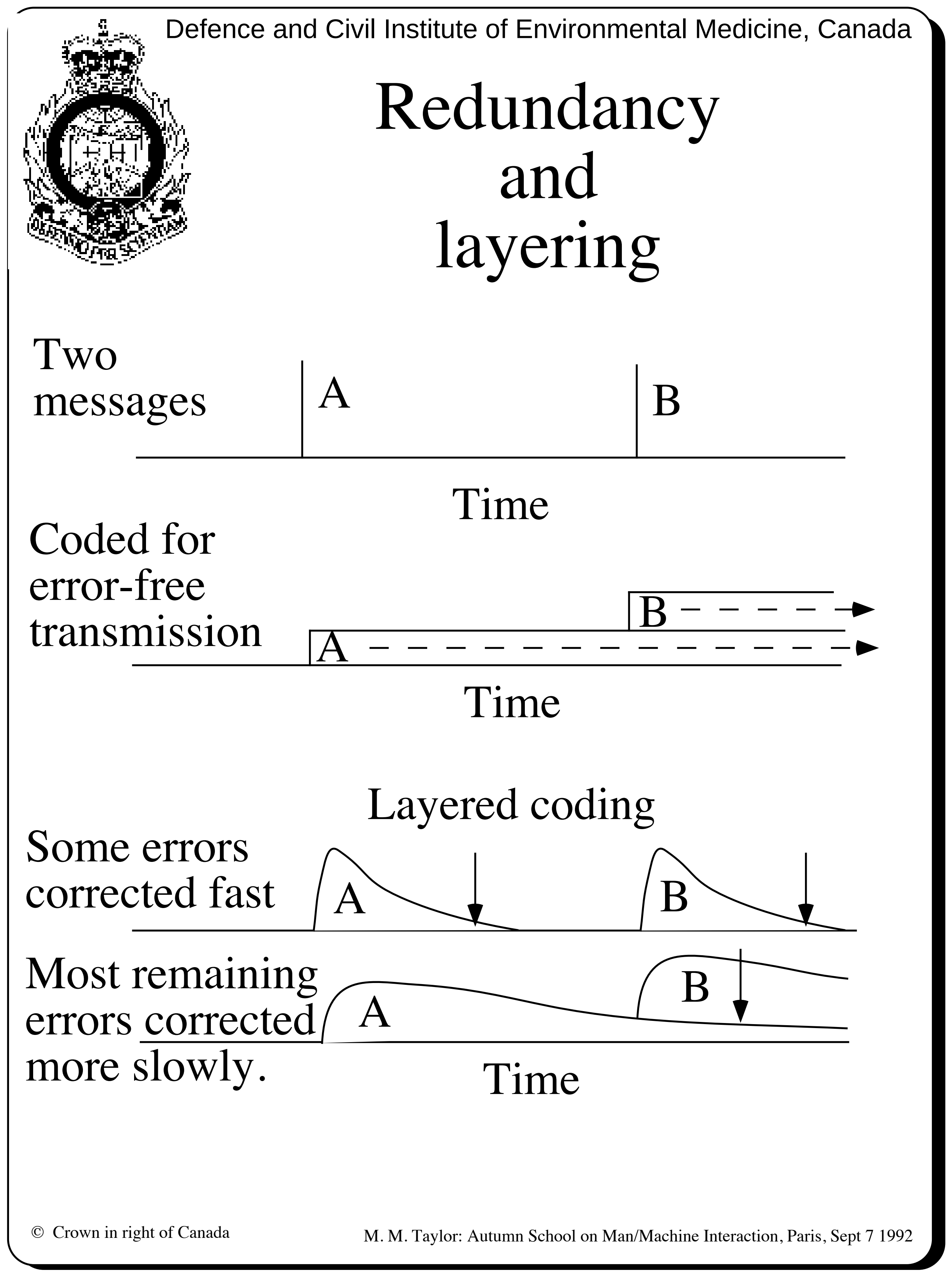
M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

This is a sketch of the interactions among the various criteria that affect stability in a feedback loop. Any specific loop can be described by a point in this space. A loop specially designed may be stable even if its point is in the region marked "Unstable Region," but a loop chosen at random is unlikely to be stable in that region.

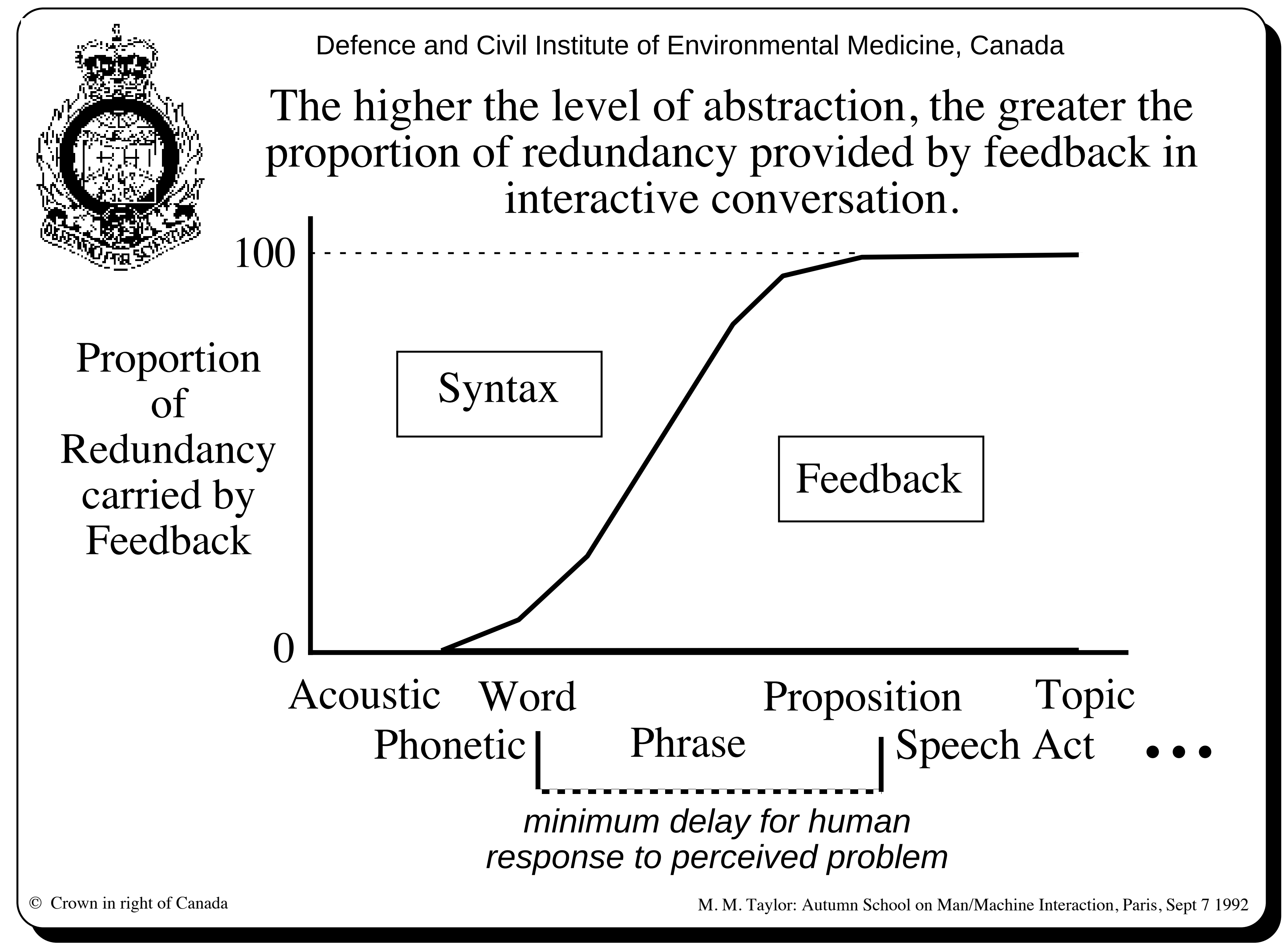
In a feedback loop, stability tends to be associated with low information rate (bandwidth, in a linear system), low gain, and low delay. The shorter the delay, the higher the safe information rate and gain. (If the gain is guaranteed to be negative, then of course higher gain makes the loop more stable; but in the generic loop, the phase of the feedback cannot be specified, so lower gain is safer than higher gain. And it is hard to ensure negative feedback with high information rate and long delay. This is especially true if the loop involves variable components, such as people.)

The external pressures on the interface are for speed and accuracy of message transmission. These pressures intrinsically conflict, but they do so most strongly if feedback is required to ensure accuracy. Time pressure demands short delay (good for stability) and high information rate (bad) which leads to high gain (bad) due to an increased need for error correction. Accuracy pressure leads to low information rate (good) but high gain (bad) due to the demand for protocol messages that ensure that the reception has been good. These demands lead to long feedback delay (bad) because of the increased time taken by the recipient to interpret the message using all available redundancy. Both pressures individually tend toward destabilizing the feedback loop, but they conflict in the way they do so. Layered protocols can help resolve the situation.

5-9



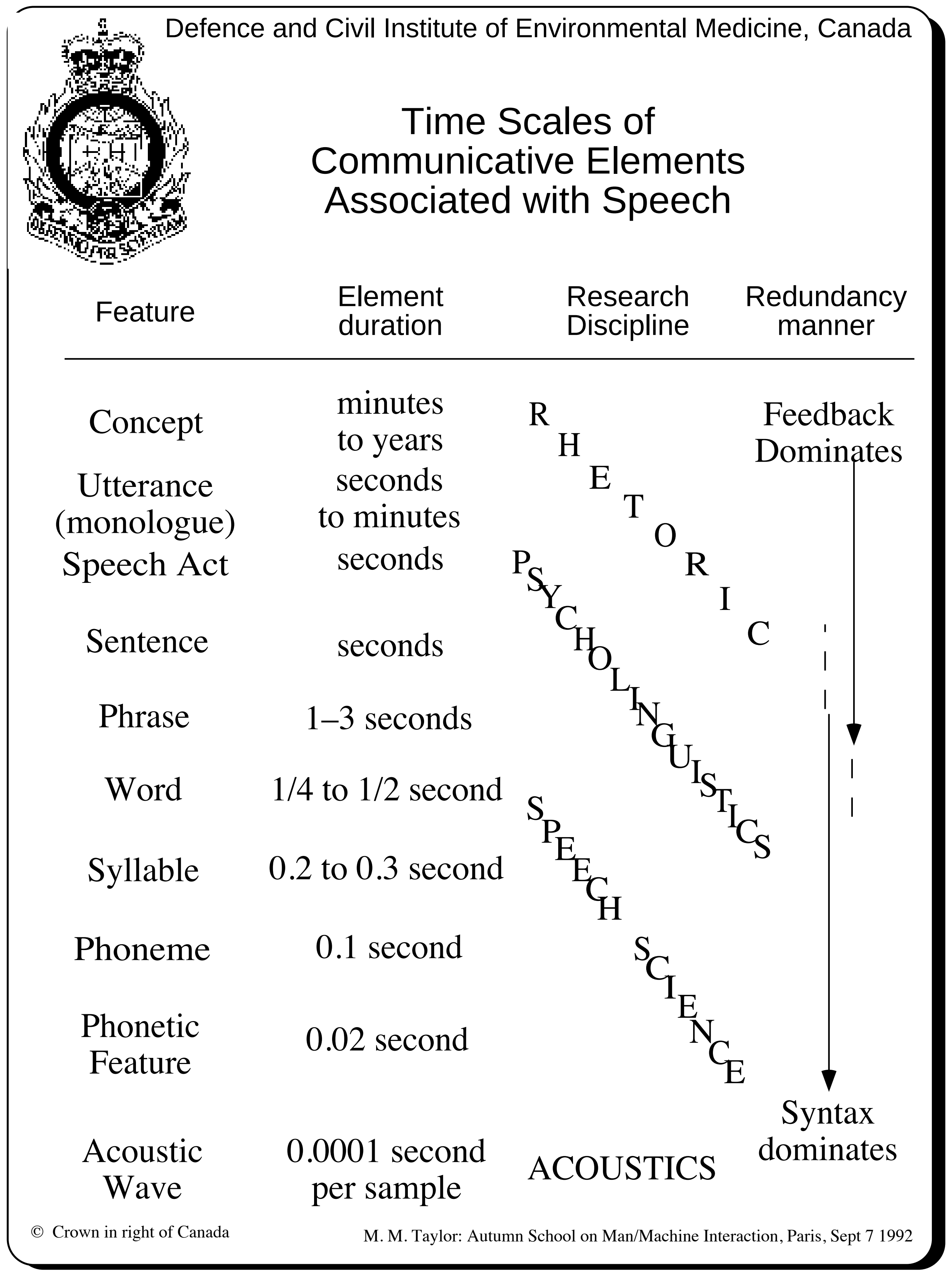
5-10



For perfect error correction the content of every message element should be dispersed over an infinite duration of the message. This is impossible, so some probability of error must be tolerated. More importantly, it would work only if the receiver had a decoder that was perfectly able to invert the encoding performed by the transmitter. That also is impossible except when both partners are computers using software protocols designed to be invertible. If there is any kind of independence between the partners, encoding-decoding cannot work, and the best that can be done is to disperse the message elements over a range of time scales, so that some errors can be corrected quickly, leaving others for slower processes.

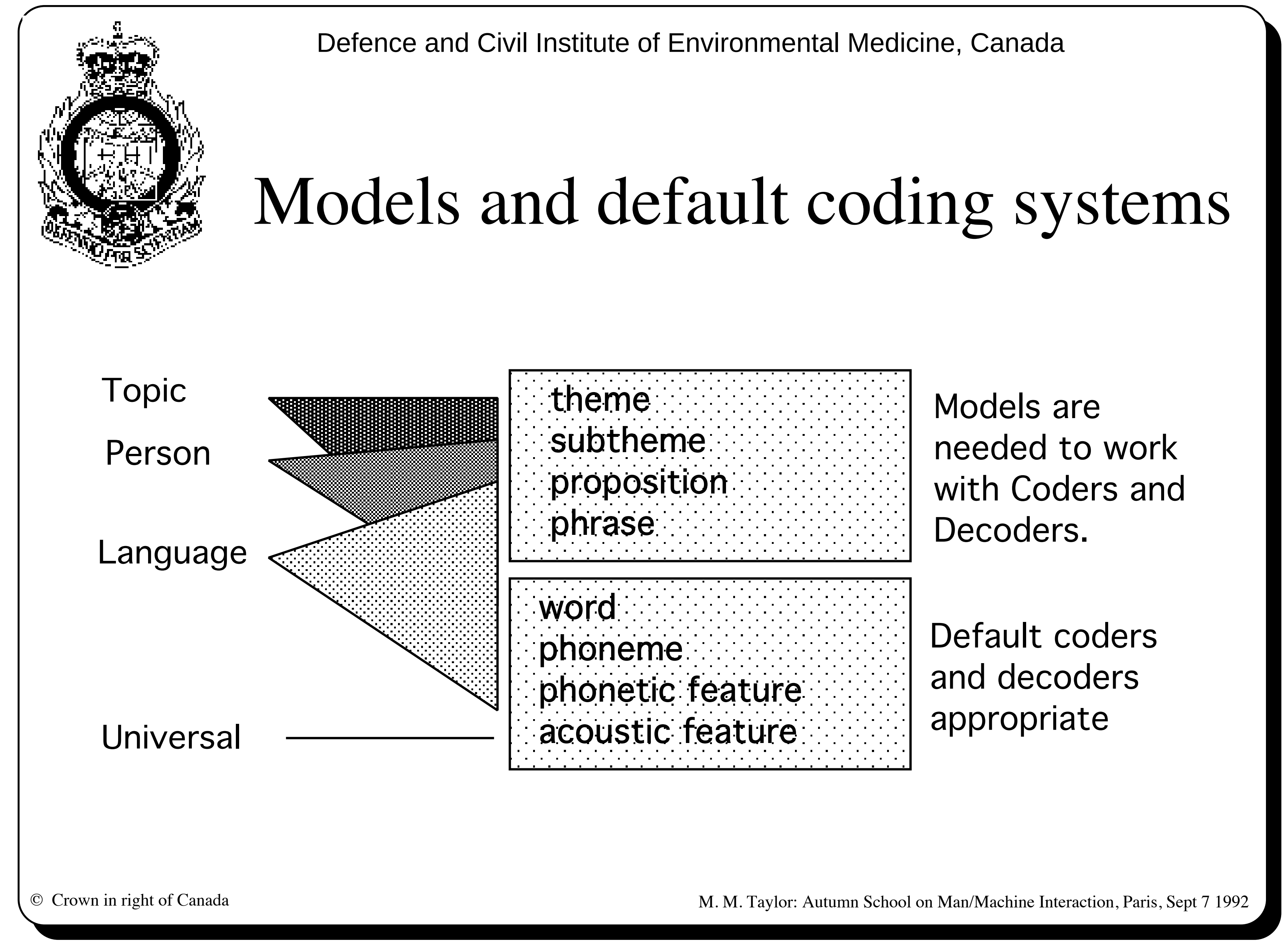
In spoken communication, several levels of abstraction are recognized by linguists. Low-level ones use a rapid succession of elements, high level ones are slower. At the lowest level, the acoustic signal, sound samples occur at several thousand per second, but there is a great deal of redundancy in converting between acoustic and phonetic representations, which occur at rates around 100 per second. These rates are much too fast for effective feedback from the partner, and all error reduction is done through redundancy (syntax) internal to the one-way message, be it acoustic, phonetic, or syllabic. But at the other end, speech acts, changes of topic, and the like occur slowly, and have very little conventionalized structure or syntax. Error correction at these levels is done almost entirely through feedback. At intermediate levels such as phrases or propositions, there is some syntax, but it is augmented through feedback and so is neither rigid nor completely specified.

5-11

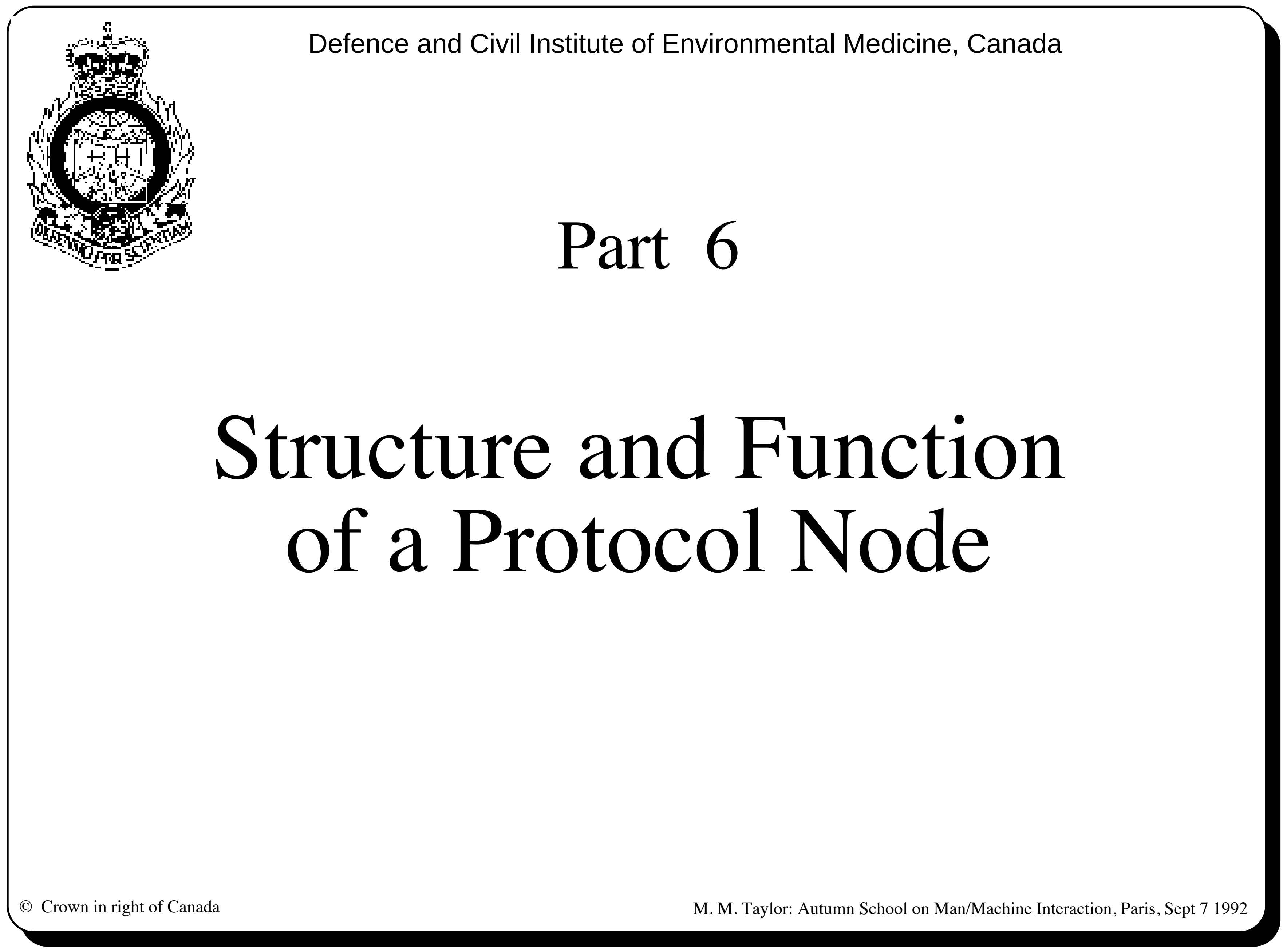


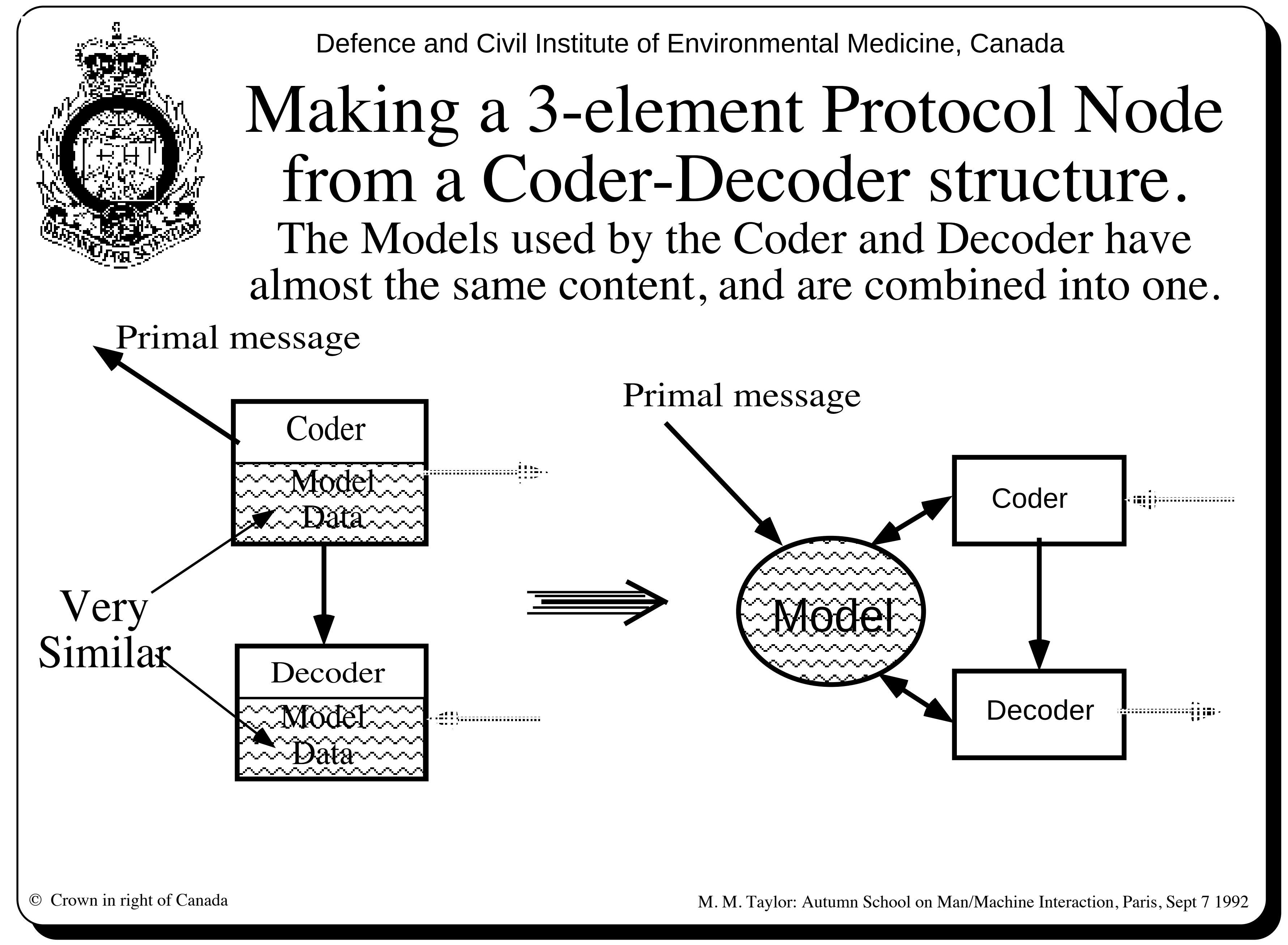
The different time scales associated with different levels of abstraction lead to different scientific disciplines concerned with them. Psycholinguistics exists in the boundary area where there is some syntax, encouraging mathematical linguists to try to describe language as if it consisted of syntax, and conversation analysts to describe it as if it had no syntax. At the lower level, speech recognition researchers try to find out exactly the form of the syntax of word construction, so that the words can be recognized, and at higher levels, rhetoricians discover how people can be influenced by modelling their reactions to language.

5-12



The more syntax there is in a level of abstraction, the more conventional must be the construction and interrelation of the message elements. Convention is equivalent to treating the recipient as being of some default type (English speaker, for example). At the lowest level, everyone that speaks uses patterns of acoustic elements to represent phonetic features, so there is universal modelling at that level. At a higher level, there are sufficiently large groups of people that use similarly related phonetic elements (and words made from them) that it is efficient to define a default model that we call a language. There are many languages in the world, but not so many as there are people. At a yet higher level, individual people have idiosyncratic ways of expressing themselves and idiosyncratic bodies of knowledge, so it is worthwhile to model individuals for communication at those levels. And at yet higher levels, the interrelation of messages depends to a great extent on the current topic or task of the communicating partners. Default models are possible for particular kinds of task, and these can be used together with the models for individuals in developing idiosyncratic conventions at the highest levels of communication. Technical or scientific jargon, or military alphabet soup, provide good examples.





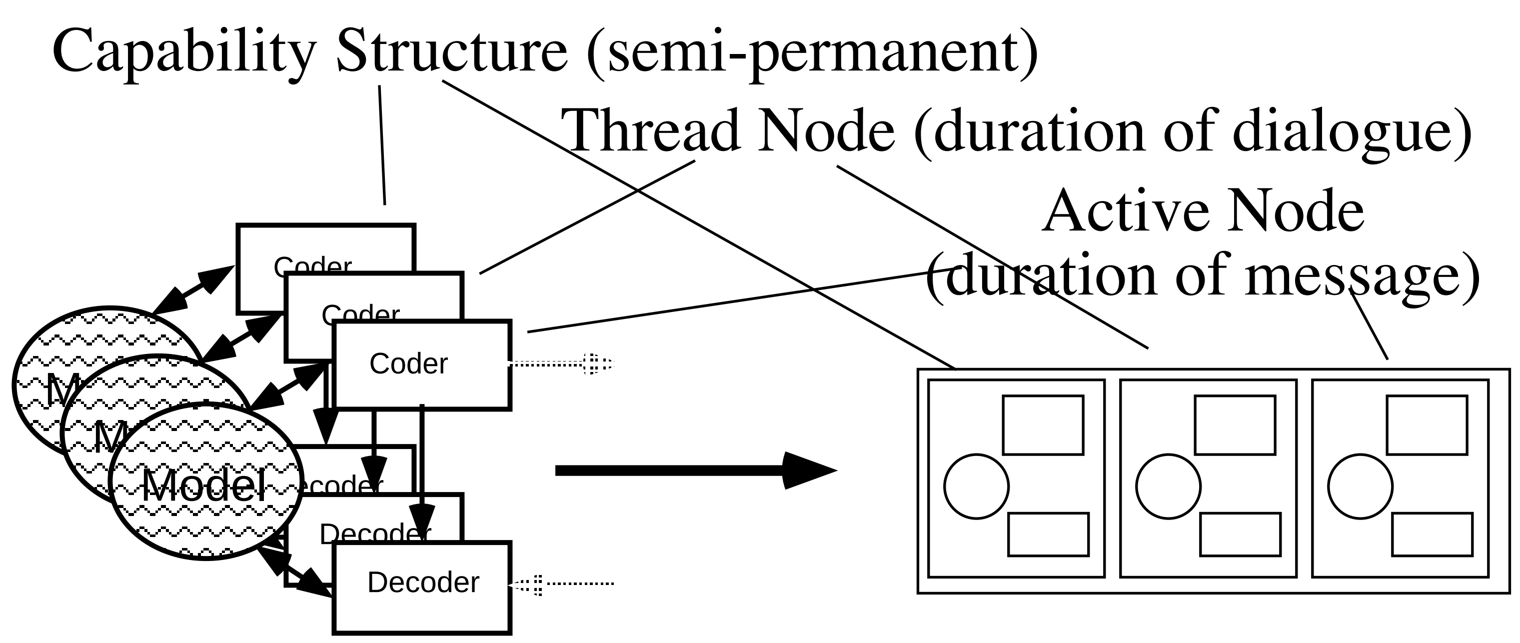
A Protocol Node (PN) is to Layered Protocol Theory as an Elementary Control System (ECS) is to Perceptual Control Theory. Seen from the outside, a PN in the Originator accepts a Primal Message that prescribes some information that the partner should have. The Primal Message is transformed into a virtual message that is sent to a PN in the recipient, and the process that performs this transformation is described as a Coder. In the recipient, a Decoder accepts the virtual message and transforms it into an interpreted Primal message, which now is the change in beliefs held by the Recipient. The Recipient encodes the necessary virtual messages that constitute the feedback by which the Originator determines the present state of the recipient's beliefs relevant to the Primal Message. A Decoder in the Originator's PN interprets these feedback messages and uses them to generate response messages that augment or modify the transmitted information. This continues until the Originator believes either that the message has been received or that it is not worth continuing the effort.

Effective transformation of the Primal Message into a form suitable for transmission to the partner involves a perception or model of the partner's beliefs about many things: the state of the world, the task, the dialogue, and not least, the originator. The model data that must be used is very similar in both the Coding and the Decoding operation, since it is against the perception of the partner's belief state that the intentions of the virtual messages can be judged.

Because of the near identity of model requirements in coding and decoding, it is convenient to distinguish a Model as a separate entity in the PN, making a "3-element Protocol Node." In this form, the analogy with the ECS becomes clearer, since it is only in the Model that the difference can be determined between the information the partner should have (the Primal Message) and the information the partner is believed to have. It is this difference that is sent as the virtual message implemented as the dialogue in this protocol.

Defence and Civil Institute of Environmental Medicine, Canada

Time Slices of the Protocol Node



 M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Defence and Civil Institute of Environmental Medicine, Canada

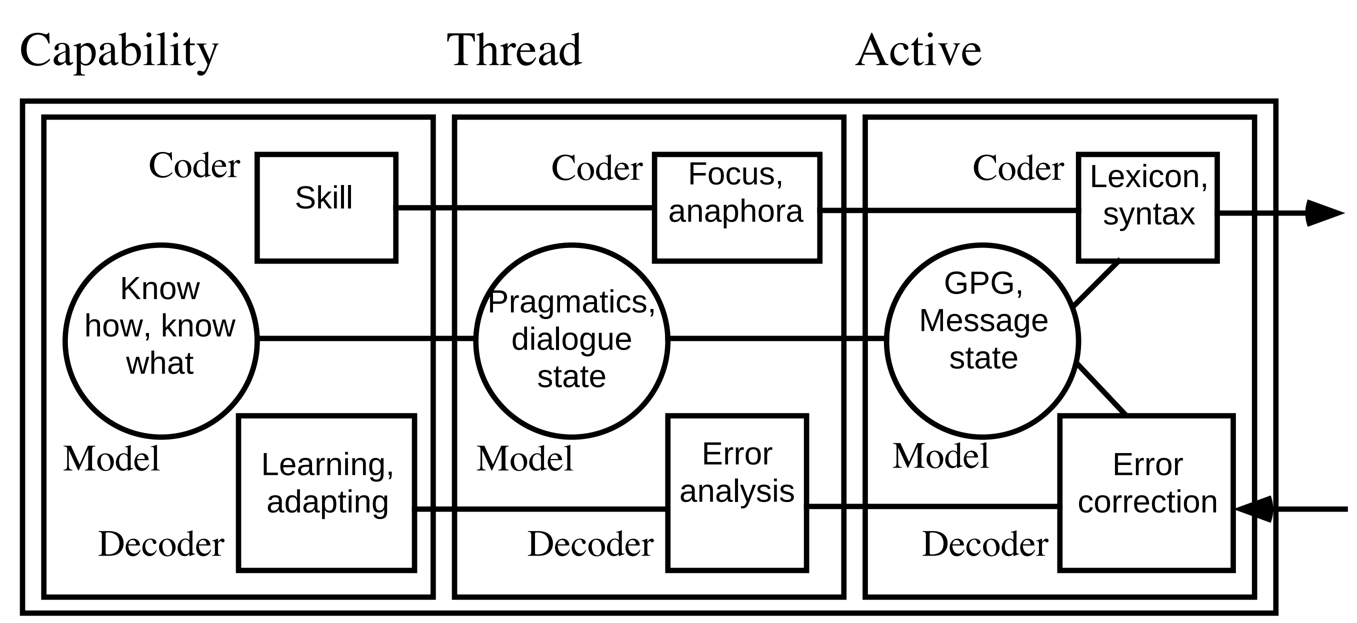
Attributes of the components of a Protocol Node

	Capability	Thread Node	Active Node
Function:	What can be en- or decoded.	What is ongoing in this dialogue.	What is active in the grammar.
Lifetime:	Long. (not permanent)	Higher request pending. Or if this the highest request, until grammar completed.	Complete pass from start to end of grammar. (one virtual message, or abort)
When Created:	At startup. (potentially dynamically)	When PN is requested.	On receipt of a new Primary for this thread.

 M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

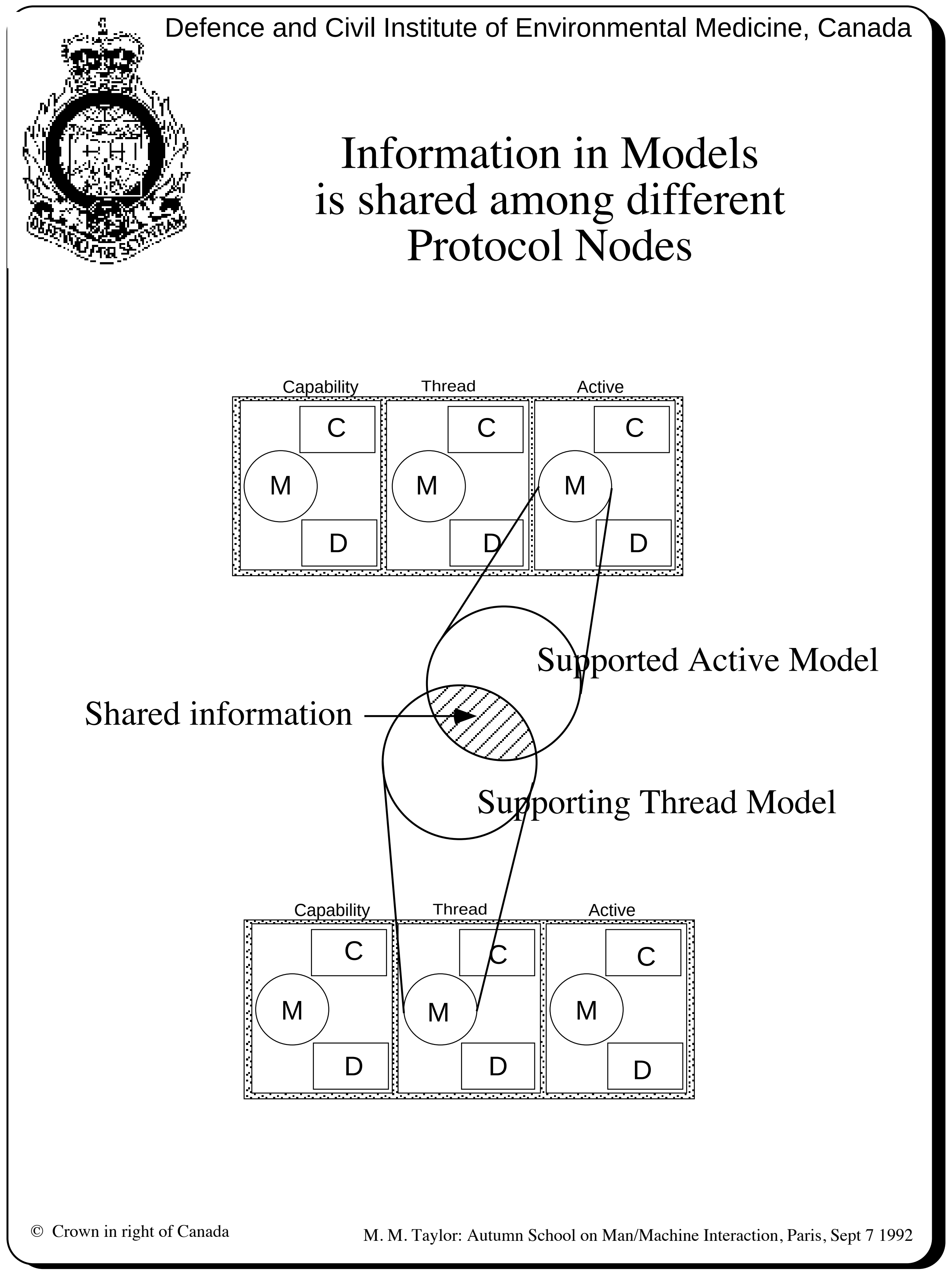
Defence and Civil Institute of Environmental Medicine, Canada

The places of some psychological and psycholinguistic phenomena within the 9-element view of a Protocol Node.

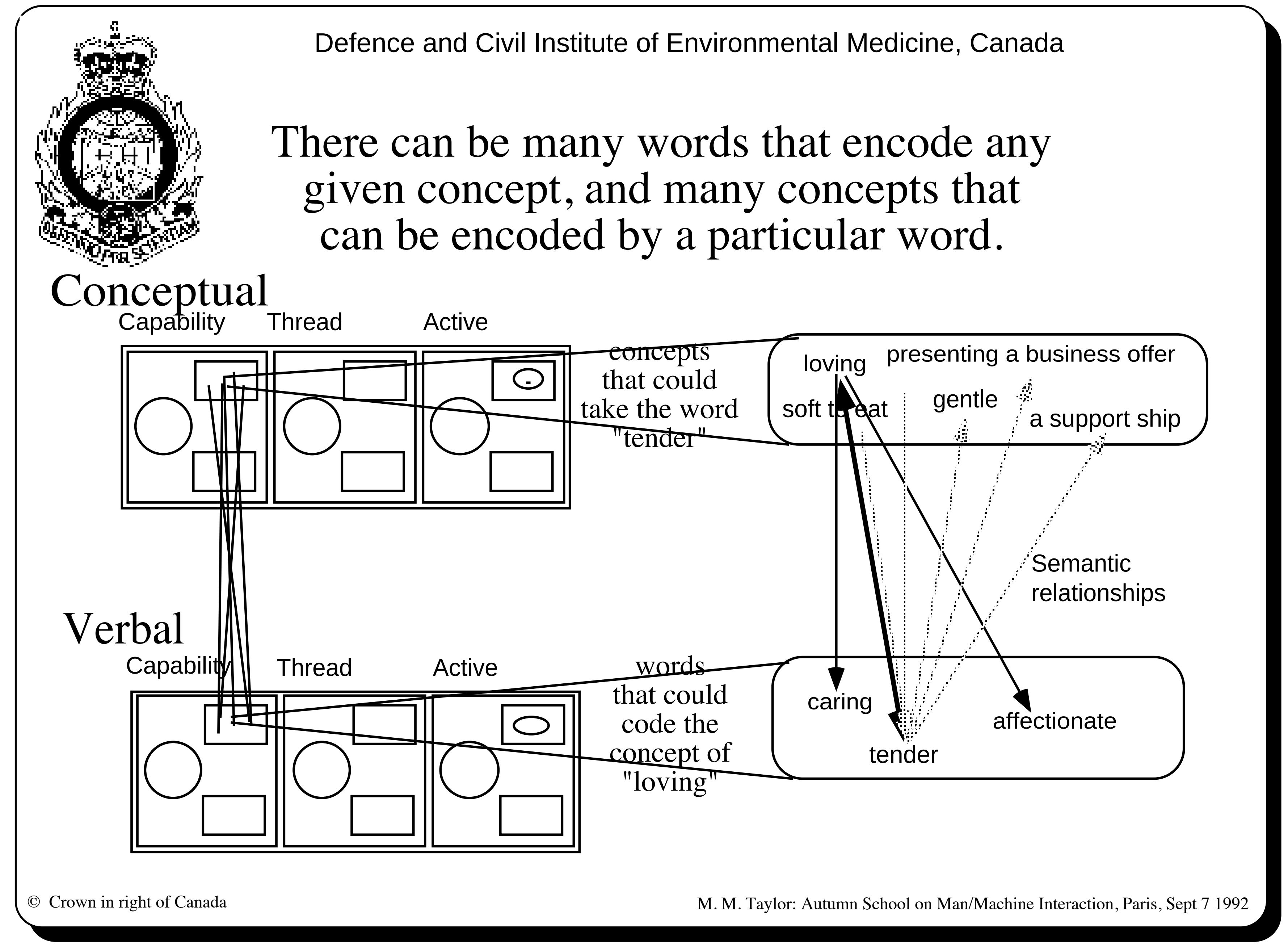


 M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

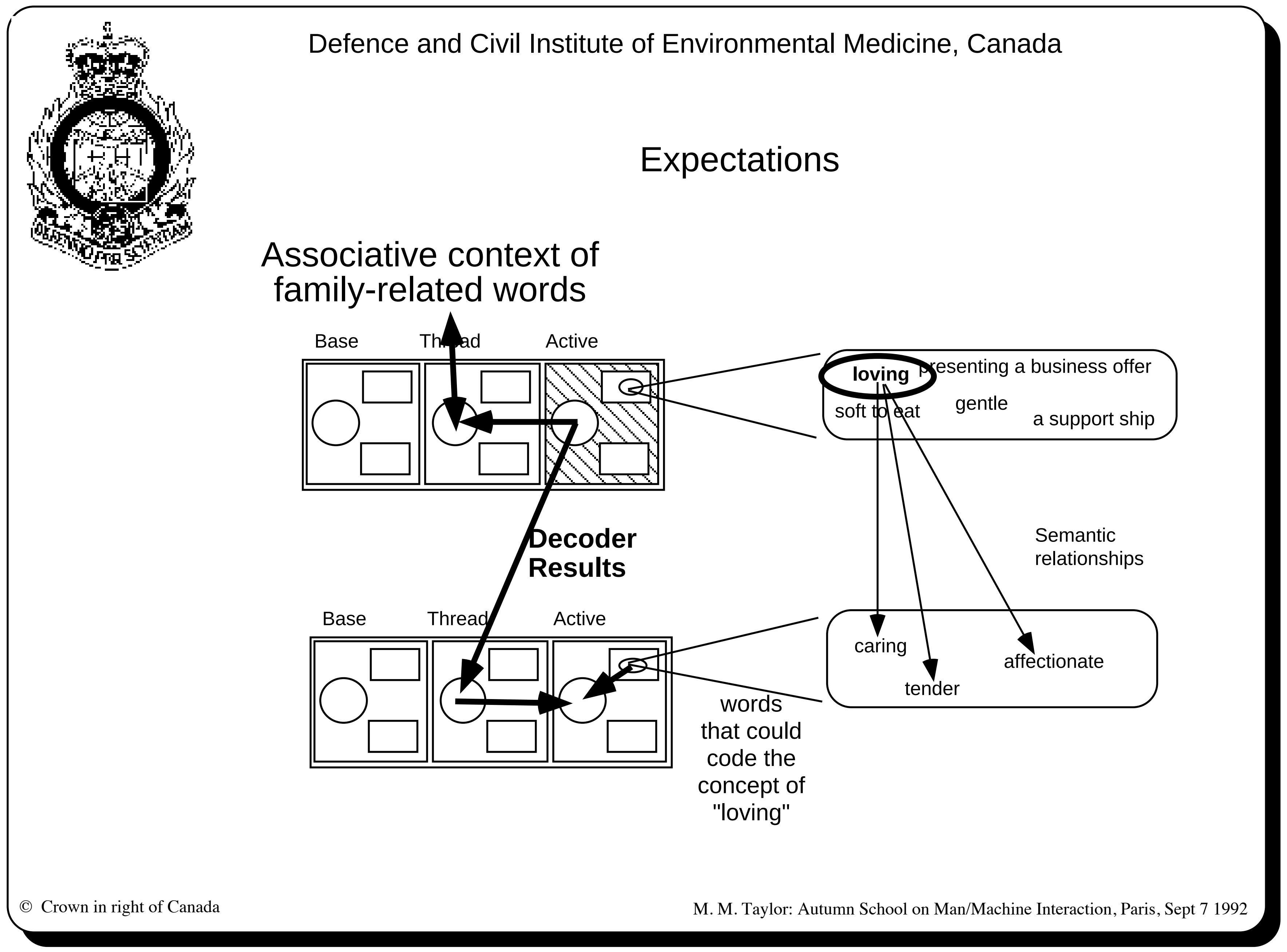
It is convenient to take the three components of a PN, and separate out three time scales over which information is valid. The longest lasting is the permanent capability, the knowledge of how to execute a protocol. It resides in what we call the "Capability Structure." The shortest time scale corresponds to the time it takes to transmit one Primal Message, including all the feedback and response messages that it takes to transmit it. Information valid only for one Primal message or less defines what we call the "Active Node." At an intermediate time scale is information relevant to this thread of dialogue, longer than one message, but shorter than the permanent capability. Such information resides in the "Thread Node." These three slices are only conceptual conveniences, and do not necessarily represent any physically separable modules. The 3 slices of the 3 components lead to the conceptual "9-element Node" that we think of as the basic structure of the PN. Each of the 9 seems easy to identify with some psychological or psycholinguistic construct.

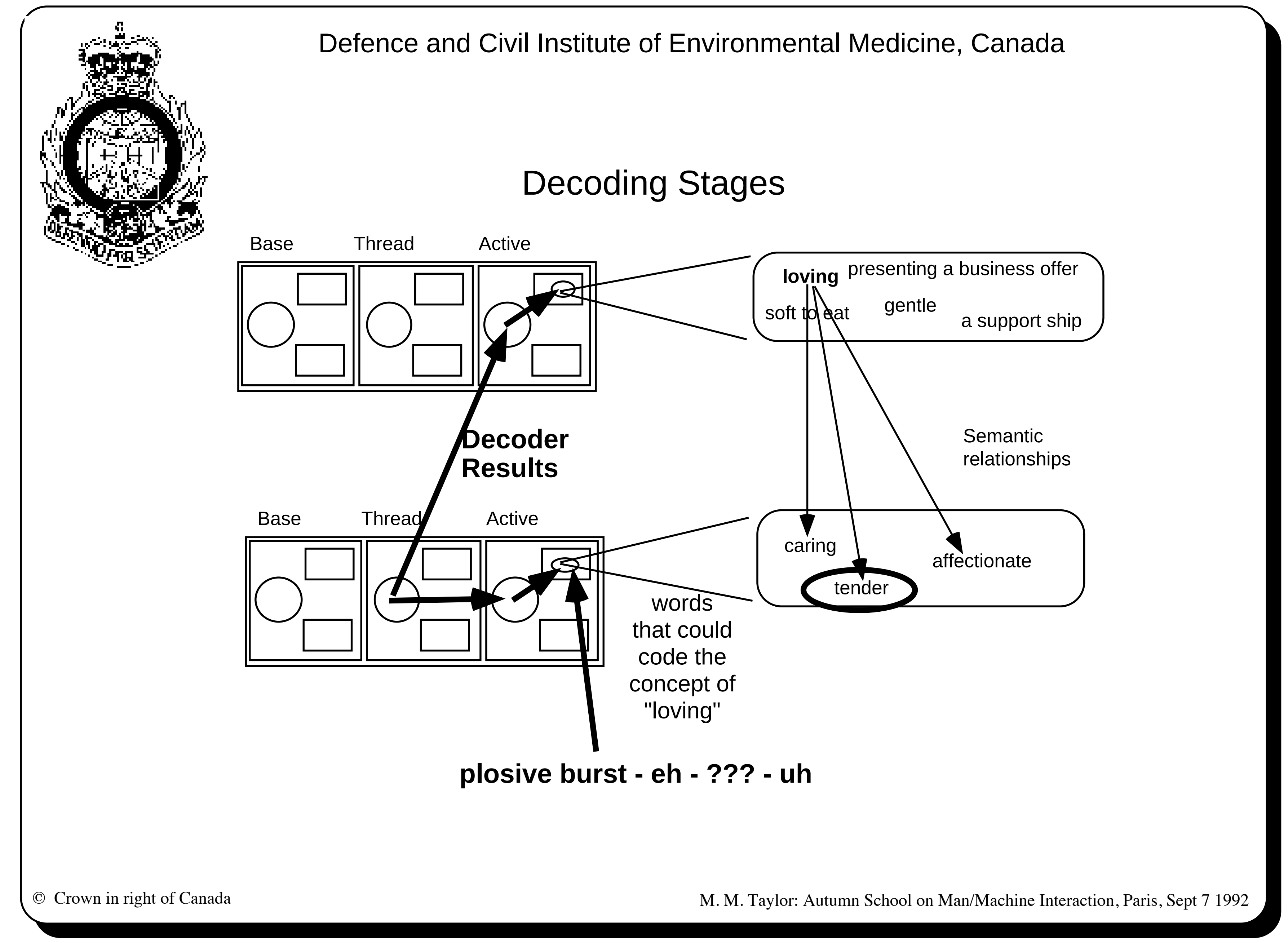


The information in the Model is not segregated, and should be conceived as being shared among the various Model elements in the different Protocol Nodes. One important linkage of this type is between the Active Model of one PN and the Thread Model of a supporting PN. The Active Model is concerned with the passage of virtual messages that contribute to the sending of the Primal Message. Each is a Primal Message in the lower PN, and therefore part of the lower Thread Model. This sharing is conceptually how the stages of abstraction relate. In an actual interface design, the linkage might well be by message-passing between the higher Active Model and the lower Thread model. Implementation is an issue separate from the conceptual structure of the inter-level relation.



Any particular concept can usually be expressed with many different words, and any one word can in most cases be used to express many different concepts. In any specific situation, only one of these links will usually be appropriate. This kind of many-to-many relationship is normal between a PN at any level and a higher-level PN that it supports. The range of possibilities that a lexical item in a lower protocol can encode in support of a higher protocol, or the inverse, the range of lower protocol choices for representing a lexical item of a higher protocol, is identified with the *semantics* of the protocol (or of the lexical item). **Semantics is a property not of any protocol, but of the relation between protocols at different layers**. In this, semantics differs from syntax and pragmatics, both of which a properties of each specific PN, disregarding completely any relationships among PNs. The semantic relationships may have the same kinds of time dependencies as do the PN properties, but we usually think of them as semi-permanent, and thus relating to the capability structures of the PNs.

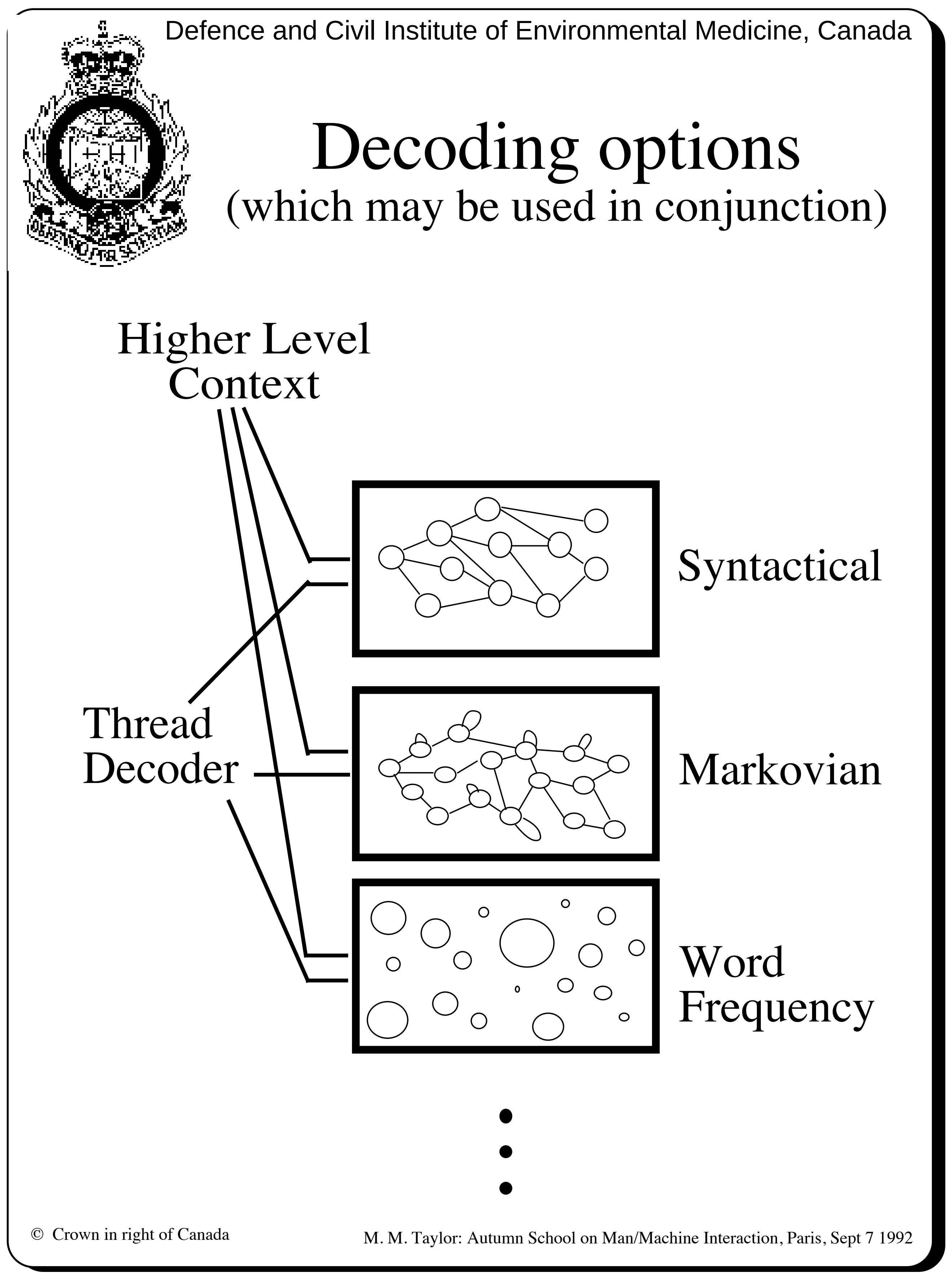




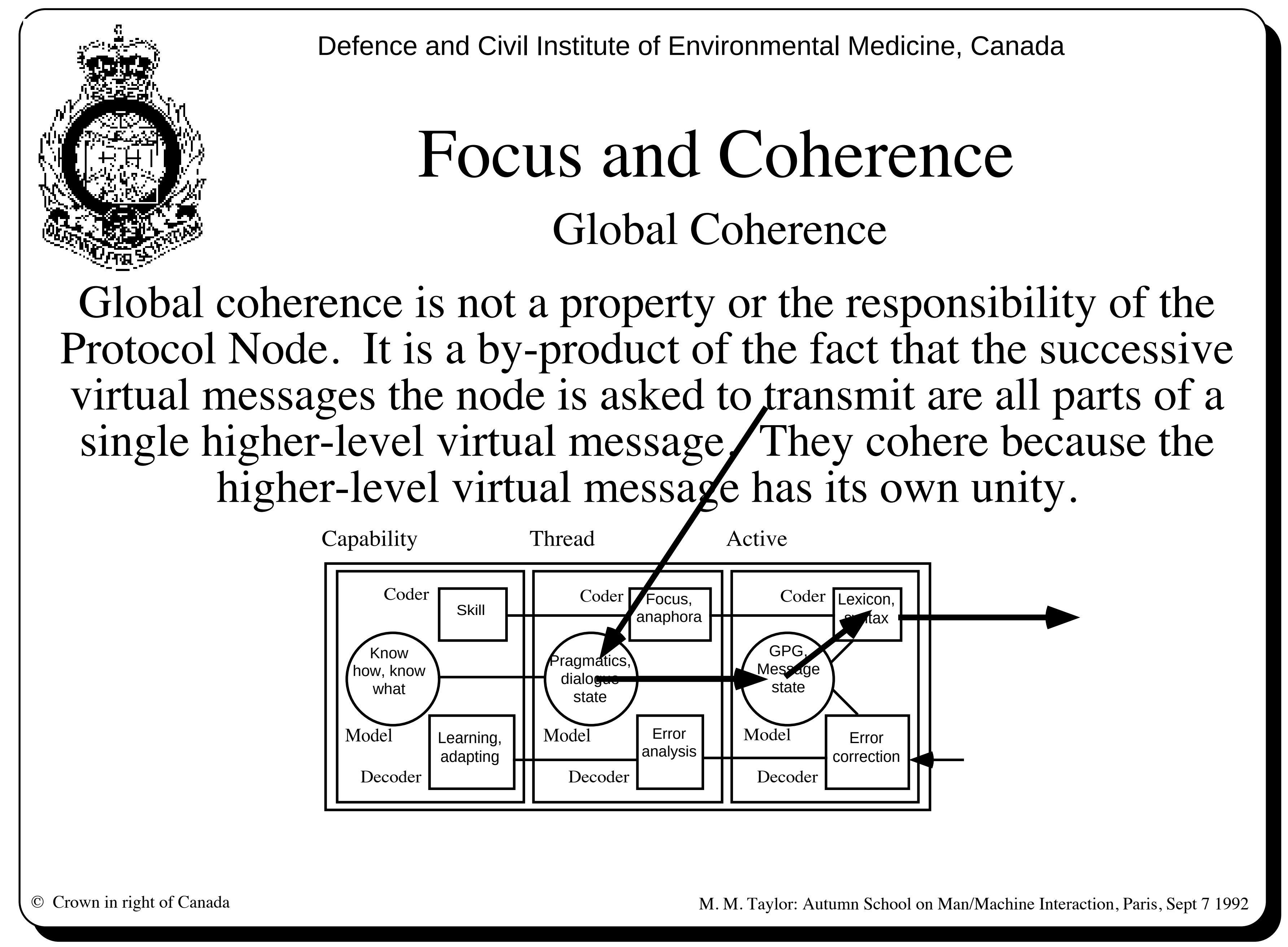
Message decoding in a protocol depends both on expectations derived from the ongoing interpretation of a higher-level message and on data derived from the interpretation of lower-level messages. In the example, a higher level message is being received that is apparently constructed in such a way that that the concept of "loving" is likely to be received (the same expectation might derive from other, pragmatic, influences on the Thread Model). This concept could be encoded by many different words, those that form its semantic range. At the word level, all of these words will tend to have an enhanced prior probability of occurrence, and thus be more readily extracted from the speech data, which are possibly noisy and inadequate themselves to distinguish among such possibilities as "tender" "fender" or "sender".

The phonetic decoding might produce something like a burst followed by an "eh" sound, something obscure, and an "uh" sound. Many words could fill this pattern, but few of them relate to the anticipated concept of "loving," or have the appropriate syntactic functions for the place in the utterance. Only "tender" seems to fit, and "tender" is what is heard..

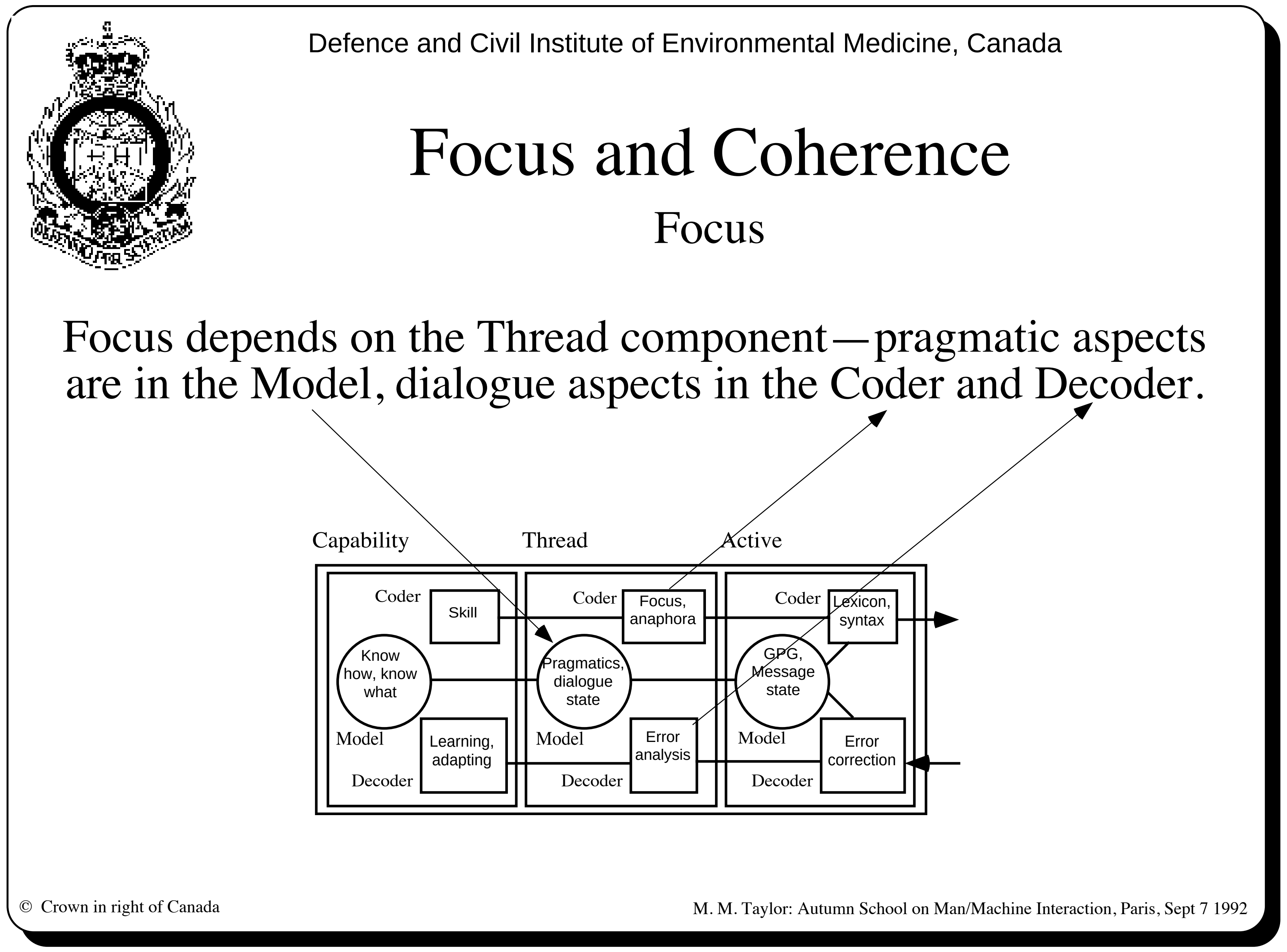
The same pattern of expectation and data enhancement of probabilities applies at higher and lower levels. If the state of an argument suggests that an explanation of some point is likely, then a new proposition is likely to be interpreted as the anticipated explanation if it makes sense to do so. Otherwise, the same word string may be taken as a new statement of fact, or to fill some other role in the argument. Misunderstandings can arise just as readily if the recipient interprets an intended explanation as a new fact as they can if a word is misheard. The principles are the same at all levels.

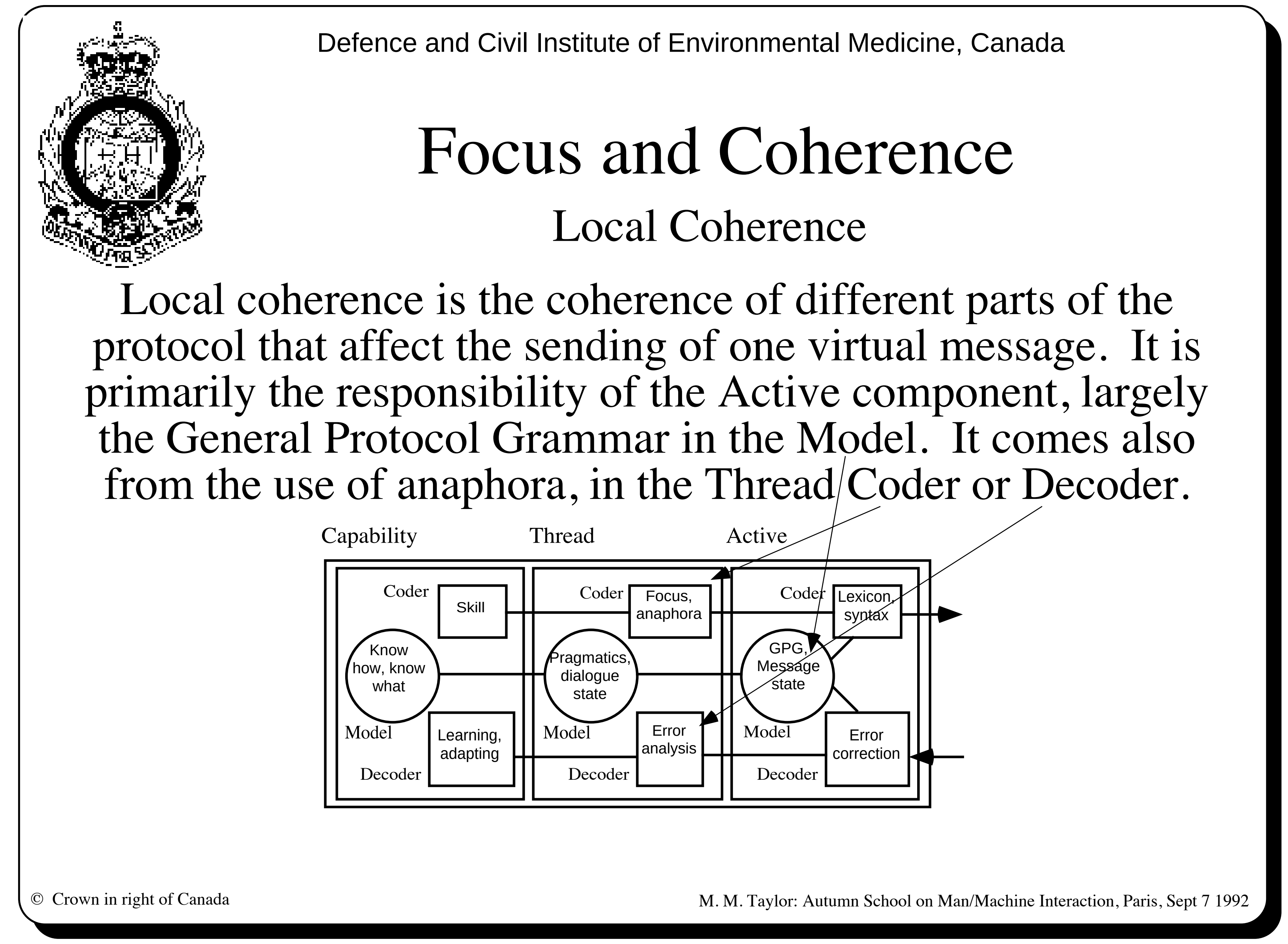


The decoder has three types of information from two sources that might assist it in making a proper interpretation of a message. The Thread component of the node contains information about the history of the dialogue to this point (and probably expectations based on convention, the pragmatic state, or past probabilities for similar dialogues). These can provide syntactical structural possibilities as well as sequentially based (Markovian) probabilities, all modulated by intrinsic probabilities of particular messages. All the probabilities are, of course, subjective, internal to the protocol node itself.



The higher level message that this protocol supports is a major source of information about what it must expect. There may be nothing in this protocol to suggest what kind of message may come next, but the internal structure of the higher message that it supports provides considerable constraint. At the word level, for example, the structure of phrases at the next level will constrain the functional class of words that could come up, and the much higher level consistency of topic will constrain the vocabulary within any open class of words. This kind of coherence can be seen by an outside observer as a consistency of word choice (at the word level, that is), but the consistency derives from sources outside the word level protocol, which knows only the sequence of Primal Messages it is being asked to deliver as individual words or word groups. *Global coherence is not a property of any protocol by itself. It is an illusion based on Local coherence at a higher protocol level.*





Focus is a matter of dialogue history and expectation, and is thus related to the Thread component of the protocol node. For each lexical class in the protocol, there could be an item that is in focus because it was the most recently used item of its class in the dialogue. This item would be maintained in the Coder or Decoder of the Thread component (depending on whether it was used by Self or Partner). It provides the target and reason for ellipsis or what we call "syntactic anaphora". The pragmatic situation may involve the Model. The affected aspects of the Model are in focus, though they may not refer directly to lexical items recently used. For example, a person might say "We've been waiting a long time," to which an appropriate reply might be "Maybe it's broken down," because the partners are at a bus stop. "Waiting" involves a pragmatic reference to a bus, which is brought into focus, even though neither the word nor the concept of bus has explicitly occurred in the dialogue.

Local coherence refers to interactions among the various elements used in the transmission of a single virtual message. It involves the syntactic structure of the message—the relative probabilities of finding particular elements in particular relationships with other elements—and the relationship among the feedback and response messages (see*the General Protocol Grammar* in Part 9 later in this talk). The selection of elements also depends on anaphora and ellipsis, which is related to Focus, and there is therefore a close relation between the concepts of local coherence and of focus.

Local coherence produces a well-structured message, Focus ensures that salient elements are not unnecessarily repeated, and Global coherence, provided from the Focus and Local coherence of higher layers, makes successive virtual messages seem to be related when within this protocol node they are not.

Defence and Civil Institute of Environmental Medicine, Canada

Part 7

More on LOCATE

What information should be in a Primal Message

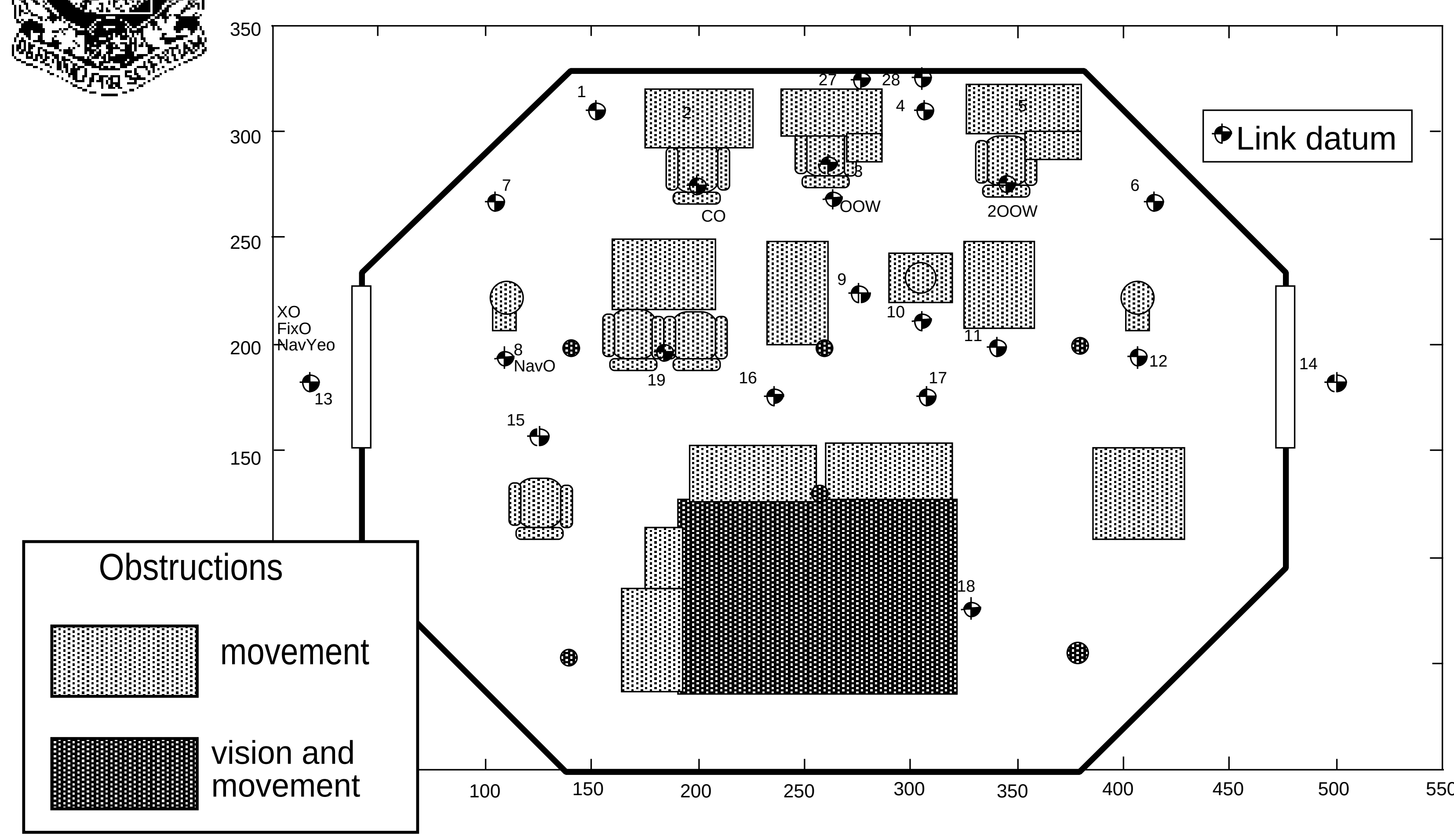
M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Part 7 is very brief. It returns to the LOCATE example, to illustrate the sort of content that might be in a Primal Message. LOCATE is not discussed in detail, and the following figures are intended only to illustrate some of the complexity of the information contained in a LOCATE workspace description. They are not intended to be examined in detail. This group of figures is followed by three that discuss what aspects of this complexity might need to be transmitted to or by the computer before the workspace can be evaluated.

The Primal Message always consists of what the Originator wants the Recipient to believe. "Believe" must be used metaphorically when the partner is a computer, but even so, the partner is expected to act in accordance with his/her/its beliefs. The interaction depends crucially on what the Originator believes about the partner's beliefs. From a third-party view, of course, there is reciprocity—the interaction also depends on what the Recipient believes about the Originator's beliefs.

Defence and Civil Institute of Environmental Medicine, Canada

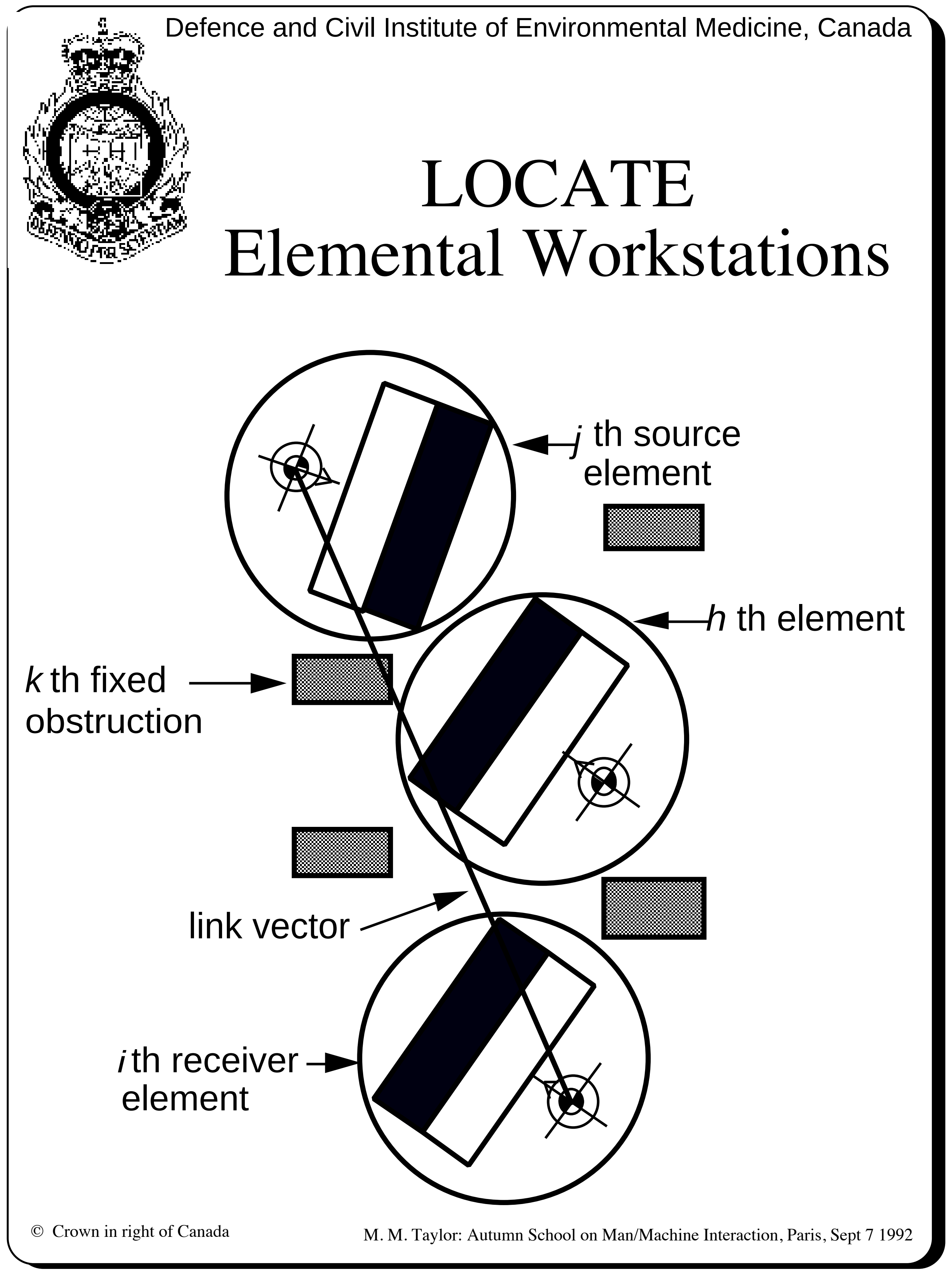
DDH280 Bridge Study



M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

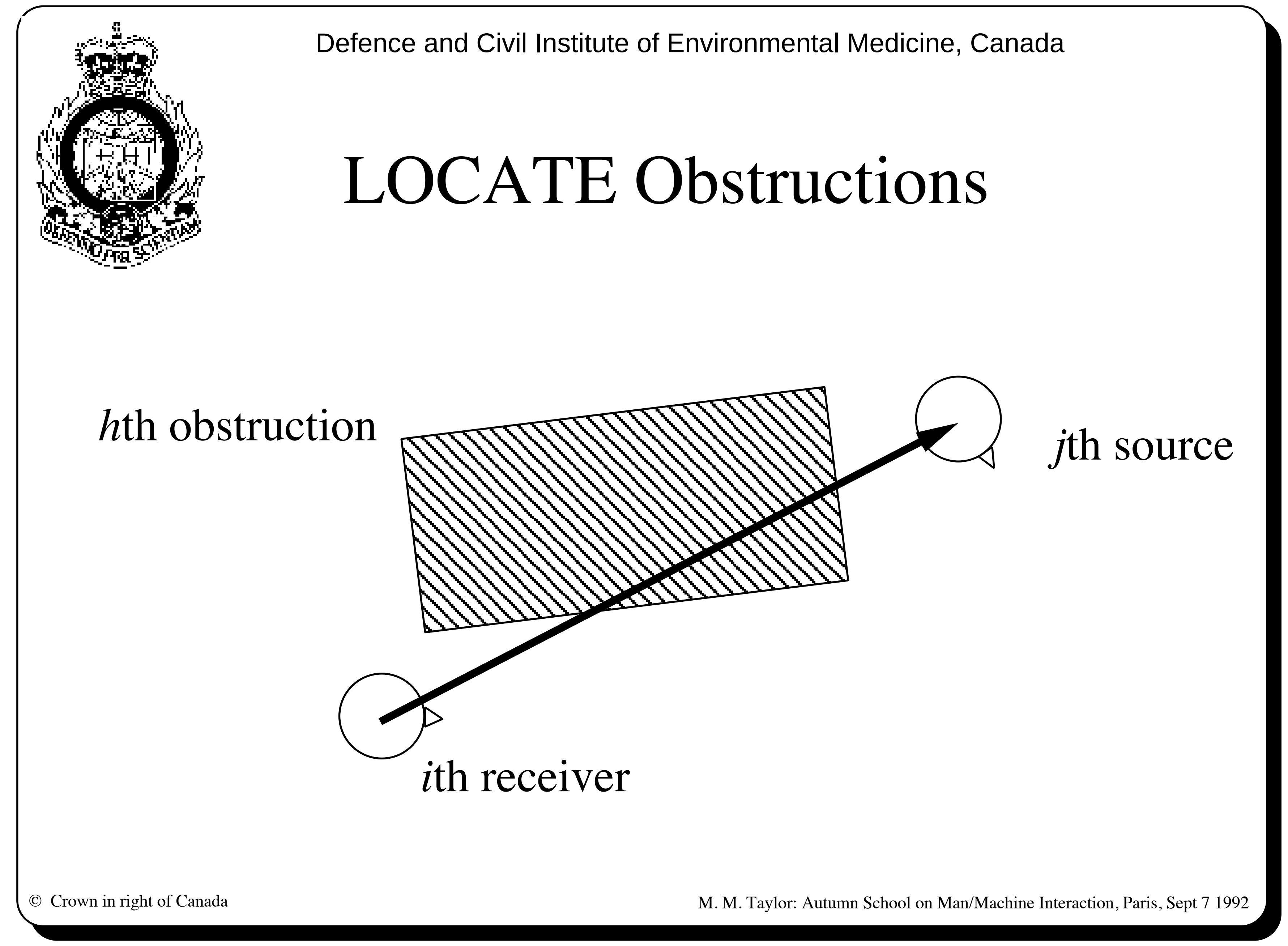
The bridge layout for a new ship provides a typical LOCATE workspace. It contains several workstations whose locations are to be determined, and several structural elements that are fixed by prior design, such as windows, columns, and walls. Workstations cannot penetrate walls, but the operators at some workstations may need to look through windows. Some elements, both fixed and relocatable, provide obstructions to vision or to motion, and there may be sources of noise (not shown) that obstruct auditory communication.

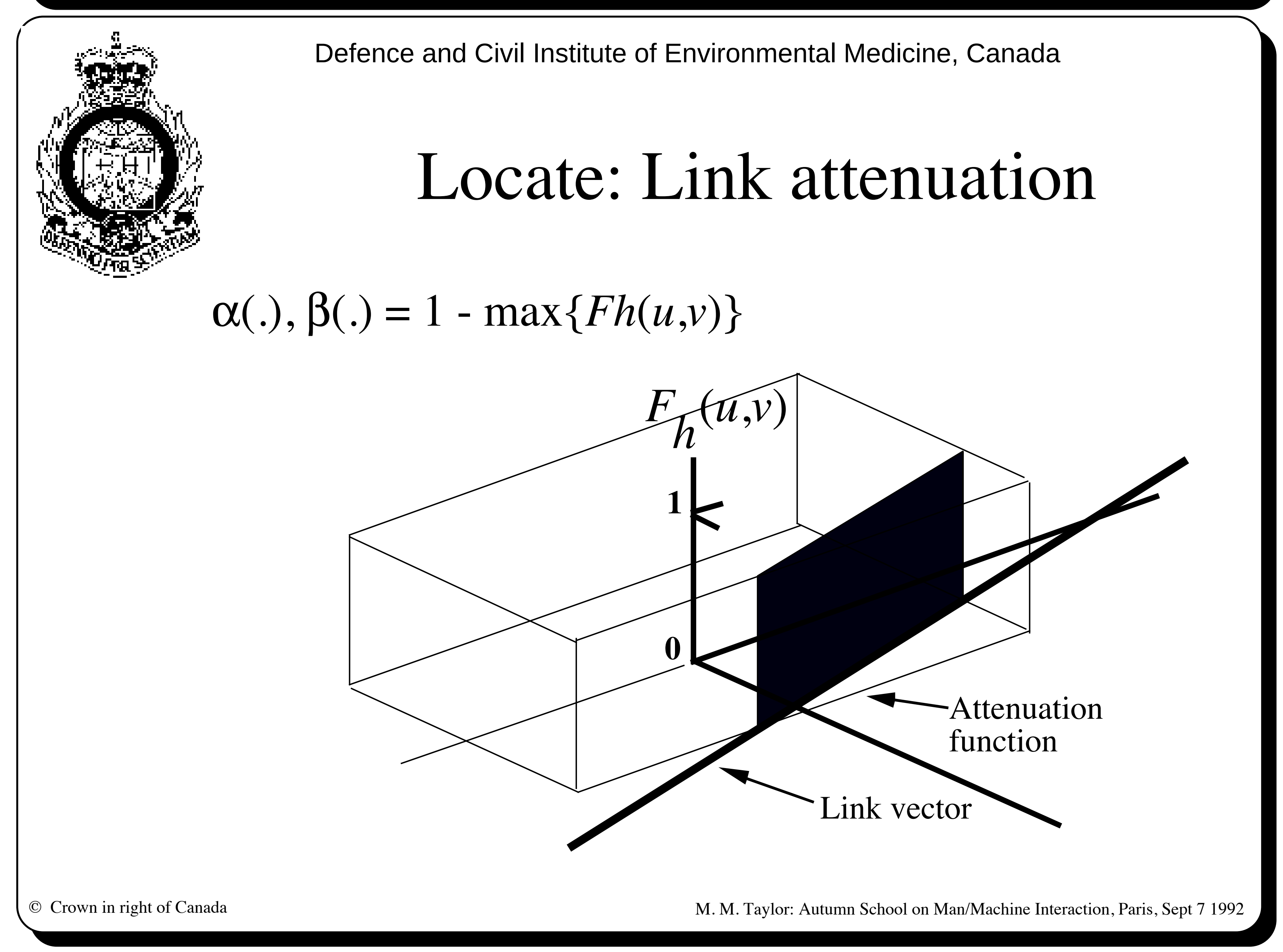
The problem is to find a layout for the workstations that optimizes the visual, auditory and other interactions among them, as well as optimizing movement requirements such as the need for an officer to move from one bridge wing (outside the area depicted) to the other. These interactions are represented by link functions that are sketched out in the next few figures.



Each workstation in a LOCATE workspace has a datum point representing the source and receiver of information. Typically it will be the operator's position. The workstation has an orientation, and it has areas that could be obstructions to interactions. Interactions are represented by link vectors connecting the datum points of the interacting workstations. Links that pass through obstructions are subject to attenuation functions (possibly, though not necessarily, infinite attenuation leading to zero interaction strength). Sources and receivers have strengths, and links have quality values as well as priorities. Domains of interaction (e.g. visual, auditory) have priorities. Each link is associated with a cost that depends on all these factors, and the evaluation of a complete layout is based on the sum of these costs.

The following figures are intended only to illustrate the complexity of the problem, and should not be studied for detailed information.





These figures are intended only to illustrate the complexity of the problem, and should not be studied for detailed information.

Attenuation Functions

For rectilinearly shaped obstructions

$$F(u,v) = \frac{1}{1 + u^{2n} + v^{2n}}.$$

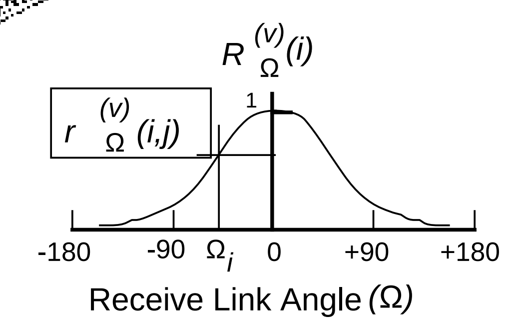
and for elliptically shaped obstructions

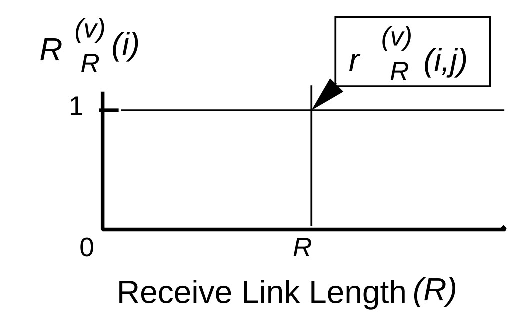
$$F(u,v) = \frac{1}{1 + \{(u^2 + v^2)/r^2\}^n}.$$

Defence and Civil Institute of Environmental Medicine, Canada

LOCATE Link Strength

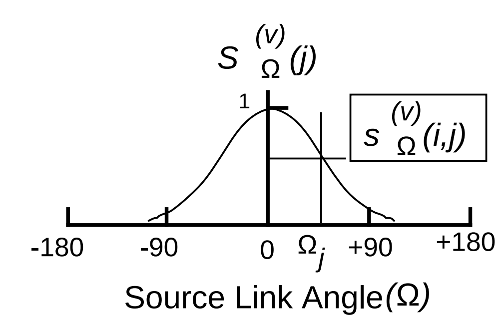
Receiver

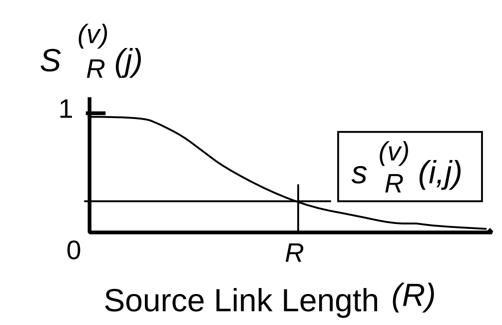




$$r^{(v)}(i,j) = r^{(v)}_{\Omega}(i,j)\ r^{(v)}_{R}(i,j)$$

Source





$$s^{(v)}(i,j) = s^{(v)}_{\Omega}(i,j)\ s^{(v)}_{R}(i,j)$$

Defence and Civil Institute of Environmental Medicine, Canada

LOCATE Link quality

Link Quality

$$q(i,j) = r(i,j)\ s(i,j) \prod \alpha(i,j,h)\ \prod \beta\,(i,j,k)$$

where

$q(i,j)$ is the *quality* of the (i,j) the link
$s(i,j)$ is the *strength* of the source information
$r(i,j)$ is the unattenuated *strength* of the received information
$\alpha(.)$ and $\beta(.)$ are *transmission* factors to account for obstructions in the workspace

Defence and Civil Institute of Environmental Medicine, Canada

LOCATE Cost Function

Composite Cost Function

$$J = \sum\sum \kappa\,\{1 - q(i,j)\}\ p(i,j)$$

where,

J is a measure of the system *cost*
$p(i,j)$ is the *priority* associated with the (i,j)th link
κ is a *weight* for each domain of communication

Defence and Civil Institute of Environmental Medicine, Canada

The Primal Message

Locate Example-1

A computable LOCATE workspace consists of a set of boundaries, a set of workstations, a set of links, and so forth. Each of these entities has attributes, such as location and orientation for workstations, strength functions and priorities for links.

In designing a workspace layout, the user believes that the computer has algorithms that might help in evaluating problems in any specific layout. But the user also believes that the computer does not know what specific layout it should evaluate. The primal message has been successfully transmitted when the user comes to believe that the computer has an acceptable workspace to evaluate.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The Primal Message consists of the information that the user wants the computer to have so that it can evaluate a workspace. A computable workspace must include all the parameters and functions described in the previous few figures, as well as the locations and orientations of some workstations. The user may model the computer as already having some of this information, by default or based on an earlier interaction. If so, the appropriate information may be referenced by, for example, naming a file containing a workspace or the description of a workstation type (analogous to the use of anaphora) rather than being presented explicitly.

Defence and Civil Institute of Environmental Medicine, Canada

The Primal Message

Locate Example-2

The top-level primal message must contain information that would permit the computer to construct an acceptable layout to evaluate. This might include, for example, that the user would find random locations for workstations acceptable, or that **this** particular workstation must be at **that** location.

If the user believes that the computer has access to the attributes of a workspace "like" the intended one, then the primal message might consist of identifying the one the computer has, and informing the computer of the differences. This is the case of access to a library or a file of saved workspaces.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

A more interesting situation arises when the user believes the computer to have information about a workspace "like" the one the user wants evaluated. The user may refer to this similar workspace and then describe modifications that bring it to the desired state. This approach is analogous to the use of **metaphor** at the interface, and is often more efficient than an explicit description. Metaphor depends on the Originator of a message believing that the Recipient can be induced to bring into focus a structure similar to the one the Originator wants the Recipient to believe, and that the Recipient can modify the basic structure retrieved from memory into the one intended by the Originator.

8-0

Defence and Civil Institute of Environmental Medicine, Canada

Part 8

Metaphor

and other aids to communication

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

8-1

Defence and Civil Institute of Environmental Medicine, Canada

Metaphor-1

A metaphor is used when

The originator wants to convey a complex concept

The originator believes the recipient knows a similar concept,

so

it is more efficient to refer to the known concept and to relate it to the new concept than to build the new concept from its conceptual elements.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Defence and Civil Institute of Environmental Medicine, Canada

Metaphor-2

Problems in using metaphor

The originator may not find a shared concept with sufficient similarity to the intended concept

The originator and recipient may have different notions of the supposedly shared complex concept

The recipient may not identify the areas of similarity intended by the originator between the shared concept and the one the originator wants to convey.

The recipient may not identify the areas of difference intended by the originator between the shared concept and the one the originator wants to convey.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Metaphors and similes may be very effective aids to communication. The Originator (O) believes the Recipient (R) to have a complex of information that shares some characteristics with the one O wants R to have. If the pattern is complex enough, it is probably more efficient for O to refer R to the information R already has and then to indicate the changes that lead to the information O wants R to have, rather than to try to develop the whole complex from its smaller building blocks. The "office" or "desktop" metaphor of the Macintosh works because the concepts of such relations as "files" that go into "folders" that can lie on a "desktop" are familiar to a large part of the target user population, and many of the modifications that are required to bring the electronic version into operation are naturally constrained by the mouse-screen context in which the metaphor is used.

Problems arise with metaphor when O and R do not share the same concepts in the way O thinks they do.

Defence and Civil Institute of Environmental Medicine, Canada

Persistent forms

A persistent form is any lasting form available to both parties. Each can assume that the other is able to refer to it. It does not decay, as the memory of an event might.

A persistent form becomes part of the model of the partner.

Anything displayed on a computer screen is a persistent form. It would be unnecessary to continue the display if the user could reliably remember the location of everything that had been displayed, even for a flash. The computer can assume that the user can locate things that remain on the screen, and the user can assume the same of the computer. The persistent display form permits selection by pointing.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The pragmatic situation affects all dialogue. What is transmitted depends on the difference between the Primal message and what the Originator believes the Recipient already to know at the level of abstraction of the protocol. If a persistent form at that level of abstraction is accessible to both, each can incorporate it into a model not only of what the other knows but also of what the other believes them to know, recursively. A persistent form can be at any level of abstraction, and can be instantiated in many ways. Consider a chess-board or a tabular list showing the state of a grand-master game. To a pair of grand-masters, either instantiation can show attacks and defences about which they can talk, but a novice may see only the lower-level forms: strangely shaped objects on a tiled plane, or a list of letters and numbers. If the persistent form is in focus for both partners at the right level of abstraction for the protocol, the Originator can be reasonably sure that references to it will be adequately interpreted.

Defence and Civil Institute of Environmental Medicine, Canada

Persistent forms and Metaphor

Persistent forms have much the same function as metaphor. Each provides a structure that the partners each believe the other can access, and that can be used for reference to other structures in the dialogue.

In using a persistent form, the originator assumes that the recipient can access it, for example to identify a picture as belonging with a certain process because it appears in a certain window on the screen.

In using metaphor, the originator assumes that the recipient can access a certain structure (e.g. that of a "desktop" or "office"), and can refer to items in it (such as a "folder" or "cabinet").

© Crown in right of Canada

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

When the Originator of a message uses metaphor, the assumption is that the Recipient's world model contains a structure that can be used for reference, in the same way that a commonly observable persistent form can be used. A reference to some aspect of the metaphorical structure can evoke much of the rest of the structure, other parts of which can be used to help link messages that would otherwise appear to lack coherence. In this, it is like a persistent form. But unlike a persistent form, the Originator cannot be assured that the Recipient's metaphorical structure is the same as the Originator's. Use of a metaphor wrongly assumed to be shared can lead to comunication errors that are hard to correct at the level at which they occur. The error is likely to be discovered at a higher level, though its source may remain mysterious. For the most part, however, metaphor and persistent form can be thought of as being much alike.

Defence and Civil Institute of Environmental Medicine, Canada

E-Feedback

Engel and Haakma
Institute for Perception Research-IPO
Eindhoven, The Netherlands

E-feedback gives the Originator some information about what messages the Recipient expects to receive.

E (Expectation)-feedback is provided by the Recipient of a message to the Originator before the Originator sends the Primary message.

E-feedback can be provided by a Persistent Form (e.g. the shape of a control and display), by a situation-dependent message (e.g. a menu display), or by a specific message suited to the occasion.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

E-feedback has many of the characteristics of a commonly available persistent form, but it is not necessarily persistent. It may be supplied by the Recipient in anticipation of the intention of the Originator to transmit a message. By providing E-feedback, the Recipient lets the Originator know of some aspect of a Model which, if shared would be likely to reduce the amount of information in the virtual message required to transmit the Primal message. Often, E-feedback allows the Originator to know what kinds of message the Recipient is predisposed to interpret and use effectively.

Engel and Haakma identify E-feedback in the shapes of equipment such as a radio (persistent E-feedback), in dynamic forms such as menu displays, and in one-shot messages. We might identify these different forms as belonging to the Capability, Thread, and Active time-scales for the protocols for which they are effective.

Defence and Civil Institute of Environmental Medicine, Canada

E-Feedback

Engel and Haakma
Institute for Perception Research-IPO
Eindhoven, The Netherlands

Example—A Menu (computer, not restaurant)

A menu provides three kinds of E-feedback.

1. Its existence tells the user that there are a limited number of messages the computer is prepared to accept.

2. Greyed-out menu items tell the user that under other circumstances there are messages available that can not now be used.

3. "Live" items tell the user the list of messages that the computer now is prepared for.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

A menu displayed by a computer may not be required by a user who is familiar with the available commands and could communicate them using a keyboard-supported protocol. But if the computer does display a menu, it has the function of a dynamic shared form, in letting the user know what the computer is in a state to do at the level of abstraction of the choices in the menu (which could be high or low). A keyboard is a kind of permanent menu that allows the user to select characters that can be used to convey word-level messages. An on-screen menu may provide choices that perform the same function as keyboard entry of words, for example in selection of an object on which to perform an already agreed function, or it might convey much more abstruse choices. But in all these cases, the computer is providing the range over which the selection of messages is acceptable to it.

Defence and Civil Institute of Environmental Medicine, Canada

Part 9

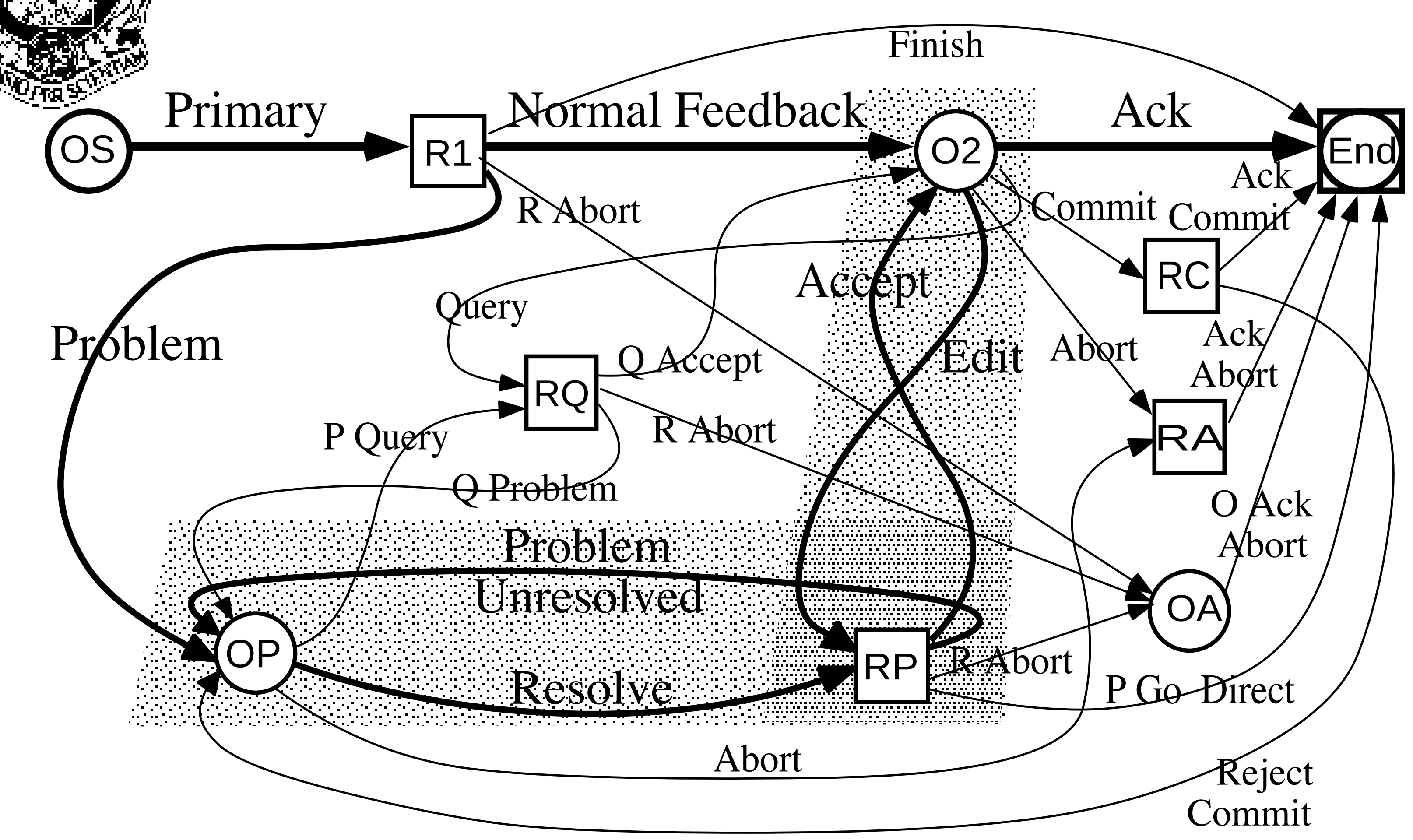
General Protocol Grammar

how messages are reliably communicated

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Defence and Civil Institute of Environmental Medicine, Canada

A GENERAL PROTOCOL GRAMMAR

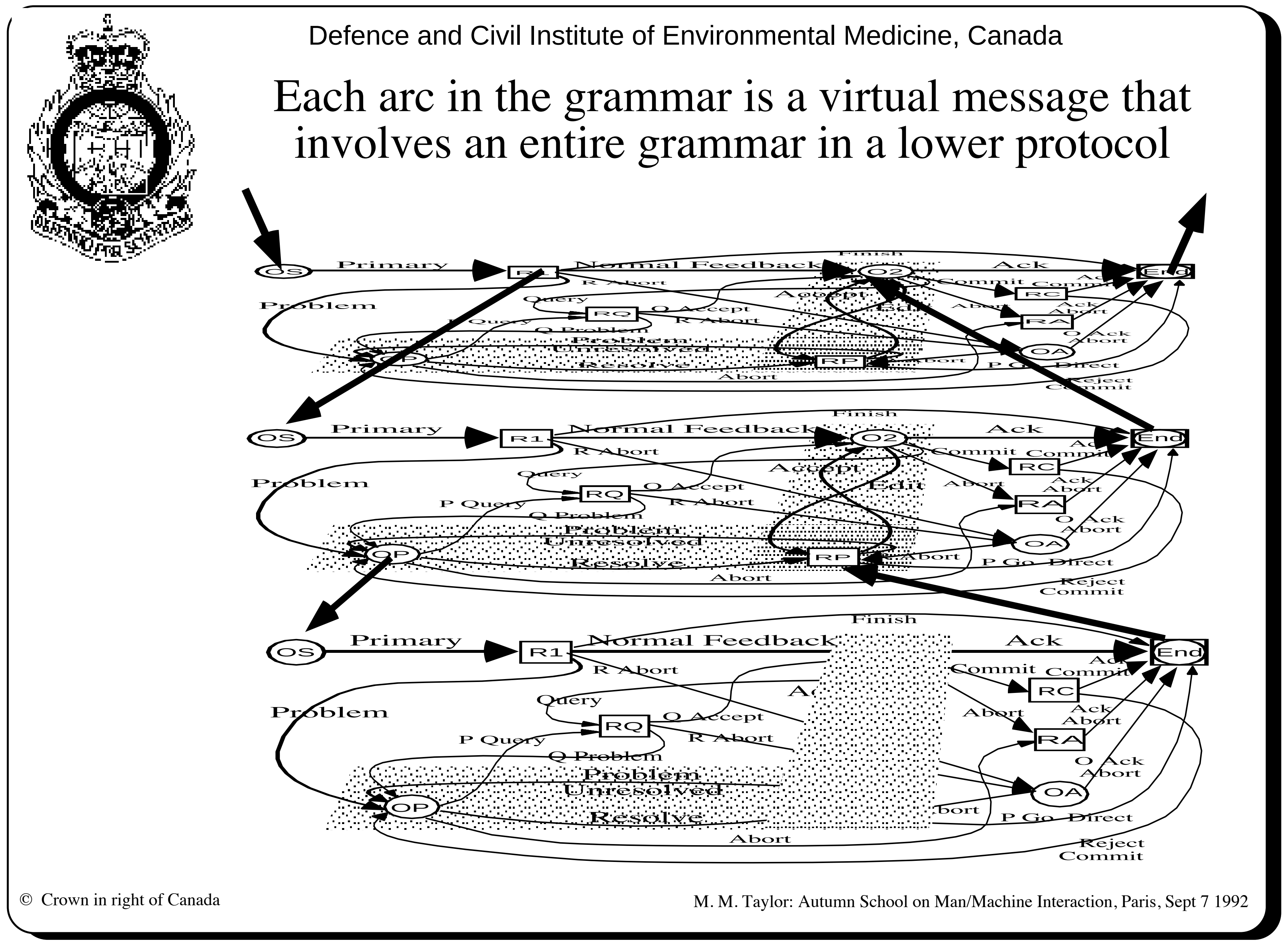


M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

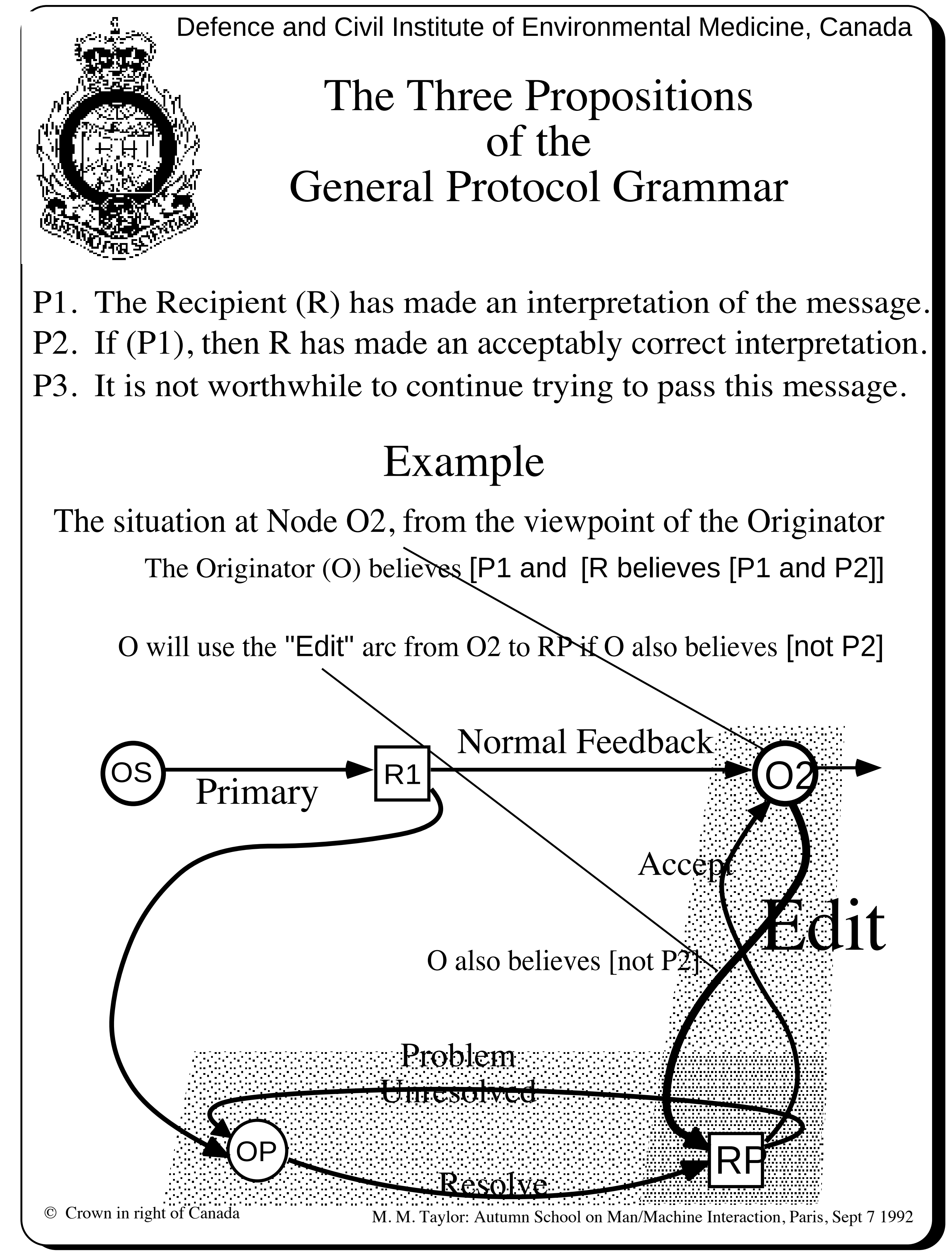
The General Protocol Grammar (GPG) describes the kinds of messages that can be passed between the two partners within any protocol. The same GPG is supposed to apply to every protocol at every level of abstraction. We describe it initially as a state transition network depicted as a set of nodes connected by directed arcs. The nodes represent states at which it is the turn of one or the other partner, the arcs the different kinds of messages that are possible.

A state transition network cannot be an accurate description of the GPG, because it requires that there be discrete transitions between states. A more accurate description involves the continuous changes in beliefs held by the two partners. The GPG diagram should be taken only as a guide to the probable patterns of belief change that may occur during the execution of a protocol.

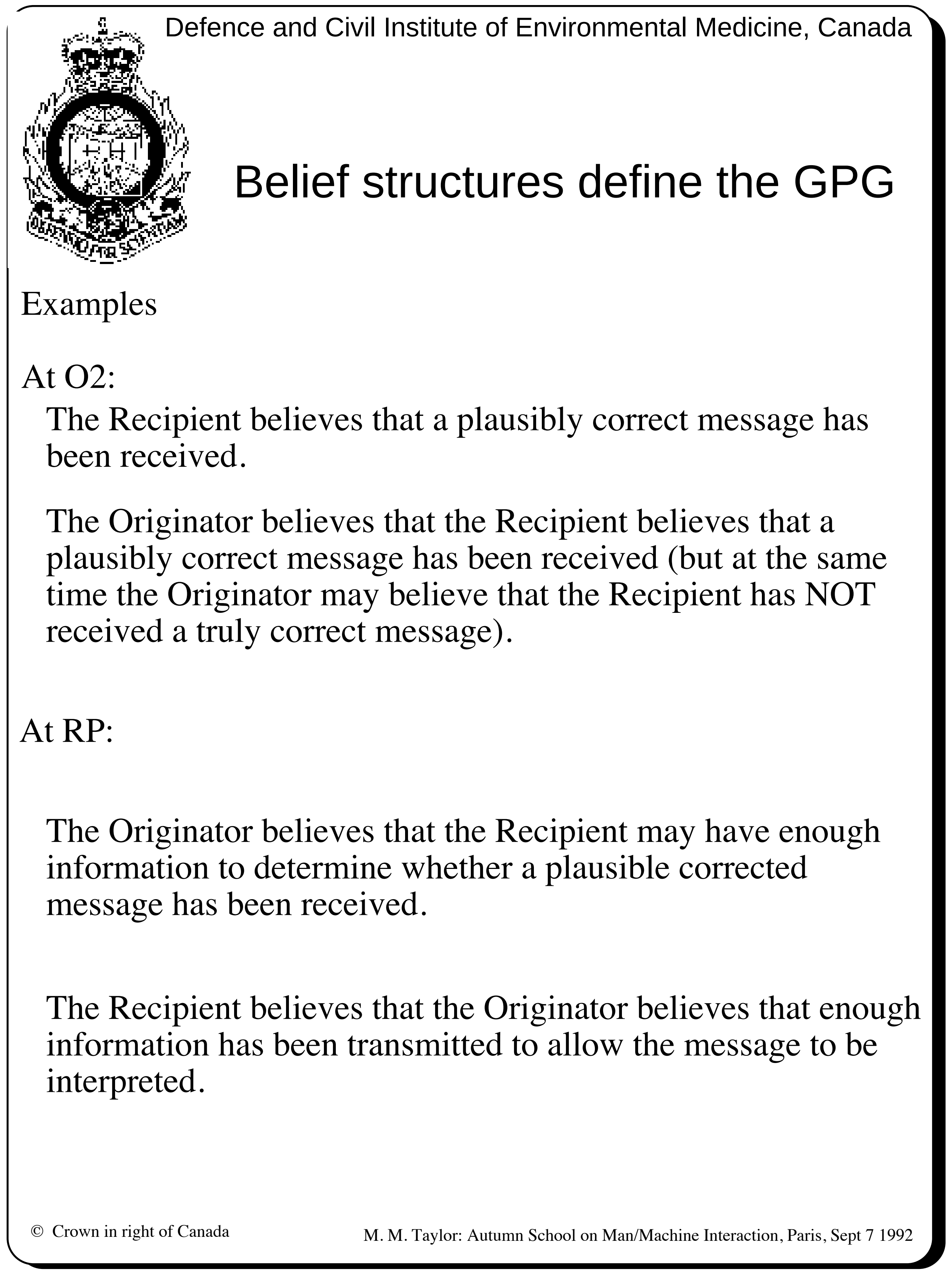
The GPG diagram shows all of the messages that are likely to occur during the transmission of a virtual message in any protocol. States at which it is the Originator's turn are shown by circles, the Recipient's turn by squares. The sending of the virtual message commences with the Primary Message, which is represented by the arc from OS to R1. At R1, the Recipient must decide whether the message as received is assuredly the one O intended to send (resulting in R taking the *Finish* arc), possibly the intended one (resulting in the *Normal Feedback* arc) or unlikely to be correct (resulting in R taking the *Problem* arc). In the diagram, commonly used arcs are shown thicker than infrequent arcs, but the frequency of use actually varies considerably with protocol level. At very low levels, there is normally no opportunity for feedback, and so a null form of the *Finish* arc is the only one used. At very high levels, it would be unusual for the initial Primary message to complete the transmission, and the partners will make much use of the arcs in the shaded areas of the diagram.



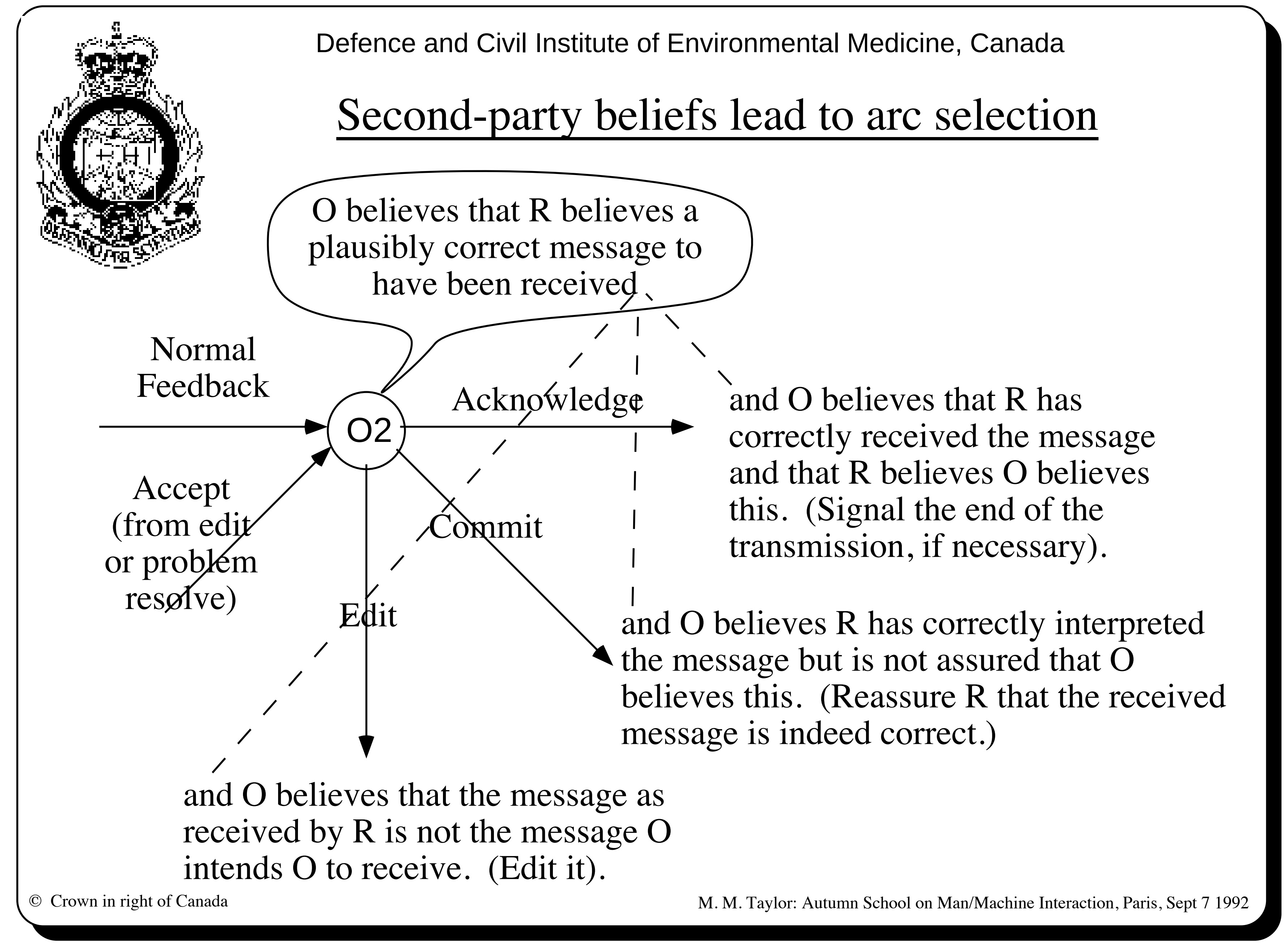
Each message, except at the physical level, is virtual, and is thus transmitted by a set of lower-level protocols. In this example, at the top depicted level, R tried to send a *Normal Feedback* message as part of the protocol for transmitting the Primal message, indicating that R was satisfied that the message received had been the one intended. But at the supporting level O did not understand this normal feedback message and took the *Problem* arc. R (who is the Originator in this supporting protocol, since it is R's Normal Feedback message that the protocol is transmitting) successfully transmitted a *Resolve* message at that level, as shown by the success of the protocol that supported it, the lowest level depicted in the figure. Eventually, O and R agree that the *Normal Feedback* message at the top level has been satisfactorily transmitted, and arrive at the top-level O2 node where it is O's turn to decide whether R has in fact received the top-level message satisfactorily, or whether it needs *Edit*ing.



The Node and Arc grammar is unsatisfactory, because it requires that the transmission of a message be completed at the lower level in order to make a discrete transition at the supported level. Calling the partners A and B, what actually happens is continuous change in the belief states one partner (say A) has about the three propositions shown in the slide, about B's belief about the propositions and about A's belief states, and about B's belief about A's beliefs about the propositions and about B's belief states about them. These three levels of recursion all affect the production of protocol messages, but we have not found it necessary to go beyond the three levels. The three levels determine which arcs in the GPG are followed, and how the followed arc is instantiated.



This figure shows some of the belief conditions that determine that the grammar is at a particular node from the point of view of one of the partners. O2 is a potential completion point that depends on the Recipient's belief in P1 & P2. If the Originator also believes P1 & P2 and that the Recipient believes P1 & P2, then the Originator will move to indicate completion of the transmission of this virtual message, perhaps by starting the next one. Similar kinds of consideration determine that the grammar is at RP. Being at RP means that O believes P1 and believes that R believes P1. O does not have a belief as to whether R believes P2, and R must indicate this by choosing the *Accept* or the *Problem Unresolved* arc.

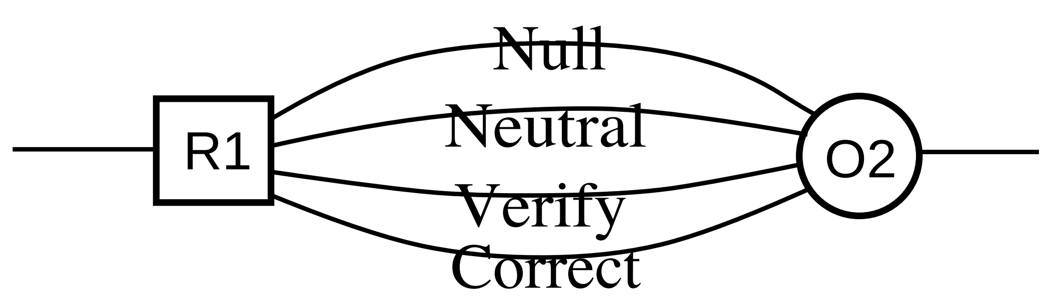


At O2, it is O's turn to send a message, which will indicate whether O believes that the message R has interpreted is the one O intended. To be at O2 implies that O believes R believes the message could plausibly be what O intended. If O agrees, it depends on O's belief as to the strength of R's belief in the correctness of interpretation whether O uses the *Acknowledge* or the *Commit* arc. If O disagrees, believing that R has not correctly and completely interpreted the intended message, O will amend the message, using the *Edit* arc. This latter is the most common case for high level messages of any complexity. It can be identified with "teaching." The choice of which arc is used to leave any node depends normally on two levels of belief recursion—belief of the turn-taker about the three propositions, and belief of the turn-taker about the belief of the other about the three propositions.

9-6

Defence and Civil Institute of Environmental Medicine, Canada

Instantiations of the Normal Feedback arc



Null	Used when R strongly believes O to expect (correctly) that R has interpreted the message as intended.
Neutral	Used when R believes that O will believe R to have correctly interpreted the message if it has been interpreted at all.
Verify	Used when R either believes weakly that the message has been correctly interpreted, or that O may not be sure as to what interpretation R has made.
Correct	Used when R believes that the message has been correctly interpreted despite a production error by O (such as a slip of the tongue), and that O is liable to make the same error again; or when R weakly believes O to have made an error but needs to check.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Most of the arcs of the GPG can be instantiated in a variety of ways. The choice of arc depends on two levels of belief recursion; the choice of instantiation often depends on a third level of belief recursion. If it is R's turn, the state depends on R's belief about the three propositions, the arc chosen by R depends on R's belief about O's belief, and the instantiation of the arc depends on R's beliefs about O's beliefs about R's beliefs. We use the very commonly used *Normal Feedback* arc as an example.

We have identified four instantiations of *Normal Feedback*, distinguished by the amount and kind of information R provides to O. Perhaps the most common instantiation is *Null*, especially at the lower protocol levels. R does absolutely nothing, because R believes O believes R got the correct message. If that is the case, R need not tell O that R believes the message was correctly interpreted, because (remember PCT) there is no discrepancy between what R wants to believe O believes and what R does believe O believes.

A very common instantiation of *Normal Feedback* is *Neutral*. R believes that O is not sure that R received the message, but that if O can be assured that R believes R received the message, then O will believe R received the correct message. In human conversation, a Neutral instantiation might take the physical form of a head tilt, a nod, a grunt, the word "Yeah," or the like.

Verify and *Correct* are less common instantiations. Their use depends on the weakness of R's belief about O's beliefs or about whether O made a mistake. *Verify* is used if R believes O believes R has received the message, but is not sure that O believes R's interpretation of the message is correct. In this case R verifies the content of the message by paraphrasing it to O. *Correct* is similar, except that in this case R believes that O made a mistake in encoding the message.

Most of the arcs in the GPG seem to have only one or two instantiations, but some have three, or as in the case of *Normal Feedback*, four. Overall, we have found about 47 different arc instantiations in the whole grammar, and this bounds the complexity of the job of the designer who must develop a particular protocol. It is important to note, however, that each instantiation is within the protocol level of the GPG, and indicates nothing about how it is encoded at lower levels. At this level, R sends, say, *Normal Feedback: Neutral*, whereas at a lower level, R may send a head nod, or may utter "yeah."

10-0

Defence and Civil Institute of Environmental Medicine, Canada

Part 10

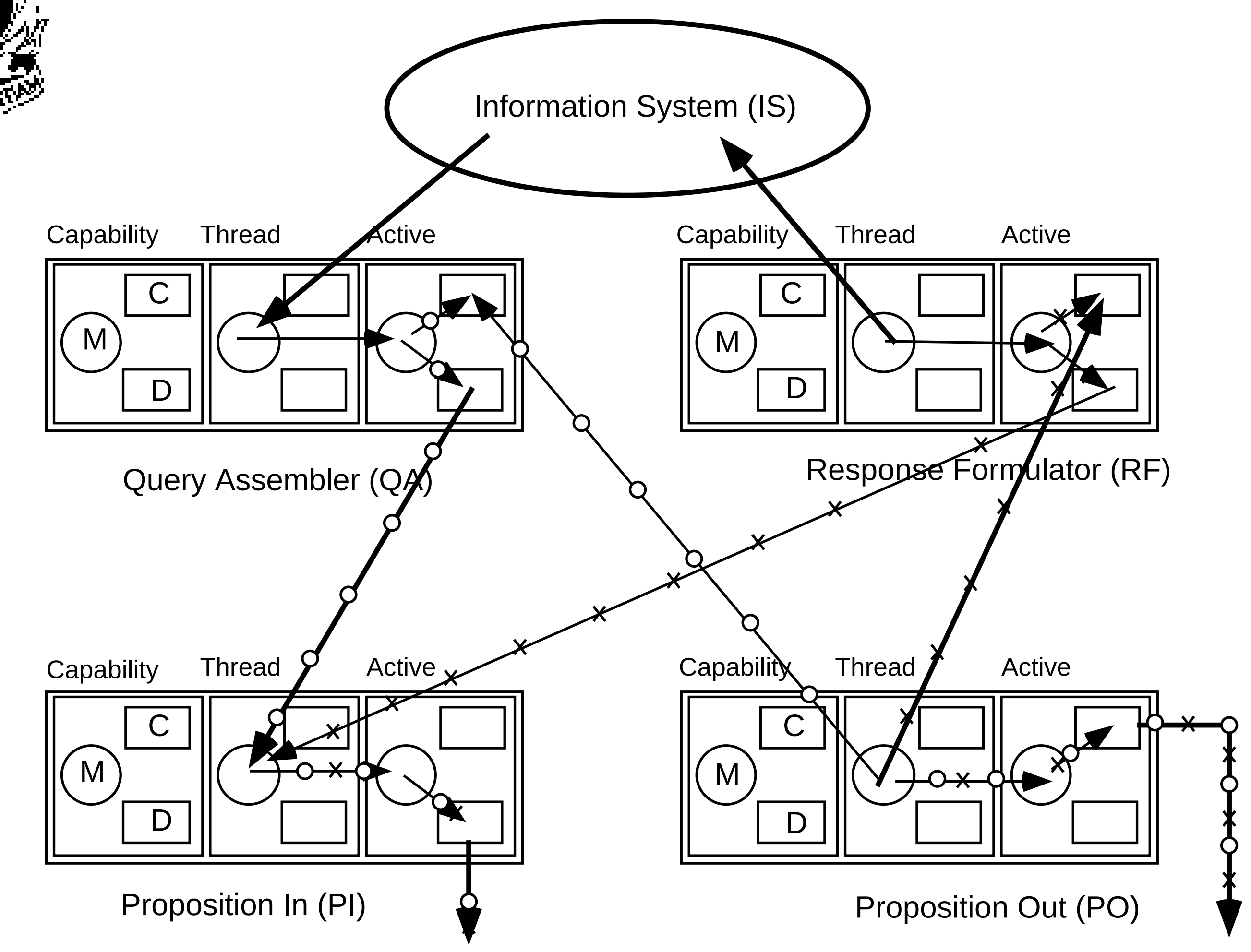
Protocols for an automated Information Service

Analysis of a simple timetable dialogue

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Defence and Civil Institute of Environmental Medicine, Canada

A simple protocol network for an information service



M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

We have used Layered protocol theory to analyze several dialogues that are supposed to simulate interactions with automated query systems of different capabilities. Here, we look only at a very simple problem in one of the dialogues. This problem causes difficulties for some analyses, but in LP theory it turns out to illustrate the distinction between syntactic anaphora (based on the most recent instance of a lexical item type) and pragmatic anaphora (based on changes in the Thread model). For an extended analysis of this dialogue, see Taylor and Waugh (in H. Bunt and W. Black (Eds.), Abduction, Belief and Context in Dialogue, Amsterdam: John Benjamin, 2000).

In our analyses of various information dialogues, we have found it necessary to propose two (and only two) protocol levels above the level of phrase. As receiving protocols (seen from the side of the pseudo-computer), we require one level that identifies simple propositions such as "I have booked flight IB885" and "Flight IB885 goes to Alicante", and another that uses the propositions to formulate queries that should be answerable by the Information System (which is assumed to be a database of some kind). As transmitting protocols, we require one that determines the information that should be provided in a response to the query, and one that forms the propositions. The formation of the linguistic output is at lower levels, not shown here. These unshown levels are used for language interactions in general, and are not specific to the information query task.

Defence and Civil Institute of Environmental Medicine, Canada

An information dialogue

(Bunt, 1989)

(W=Wizard; C=Client)

W1: Schiphol information.
C1: I have booked for flight IB 885, next Saturday, to Alicante. What time should I report at Schiphol?
W2: You should check in half an hour before departures at the latest.
C2: So between what time and what time?
W3: Between twelve and twelve-thirty.
C3: Do you also have information about departure and arrival times of trains?
W4: In Holland?
C4: Yes.
W5: I do.
A problem
C5: What is the last train from Breda I can take to be in time for flight IB 885?
W6: The train of 12.06.
C6: What is the arrival time in Alicante?
W7: 17.00
C7: What is the duration of the bus travel Alicante Benidorm?
W8: We don't have information about that.
C8: Thank you.
W9: You're welcome.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The dialogue was collected using a Wizard of Oz technique (a human pretends to be an automated system) by Bunt (1989). The client starts by providing information that allows the Wizard to interpret the question that is to come (this is a form of E-feedback, as defined by Engel and Haakma—see Part 8 of this talk). Then the initial question is asked, provoking a problem (at C2) that we analyze elsewhere (see Taylor and Waugh in Bunt and Black (Eds.), 2000).

Defence and Civil Institute of Environmental Medicine, Canada

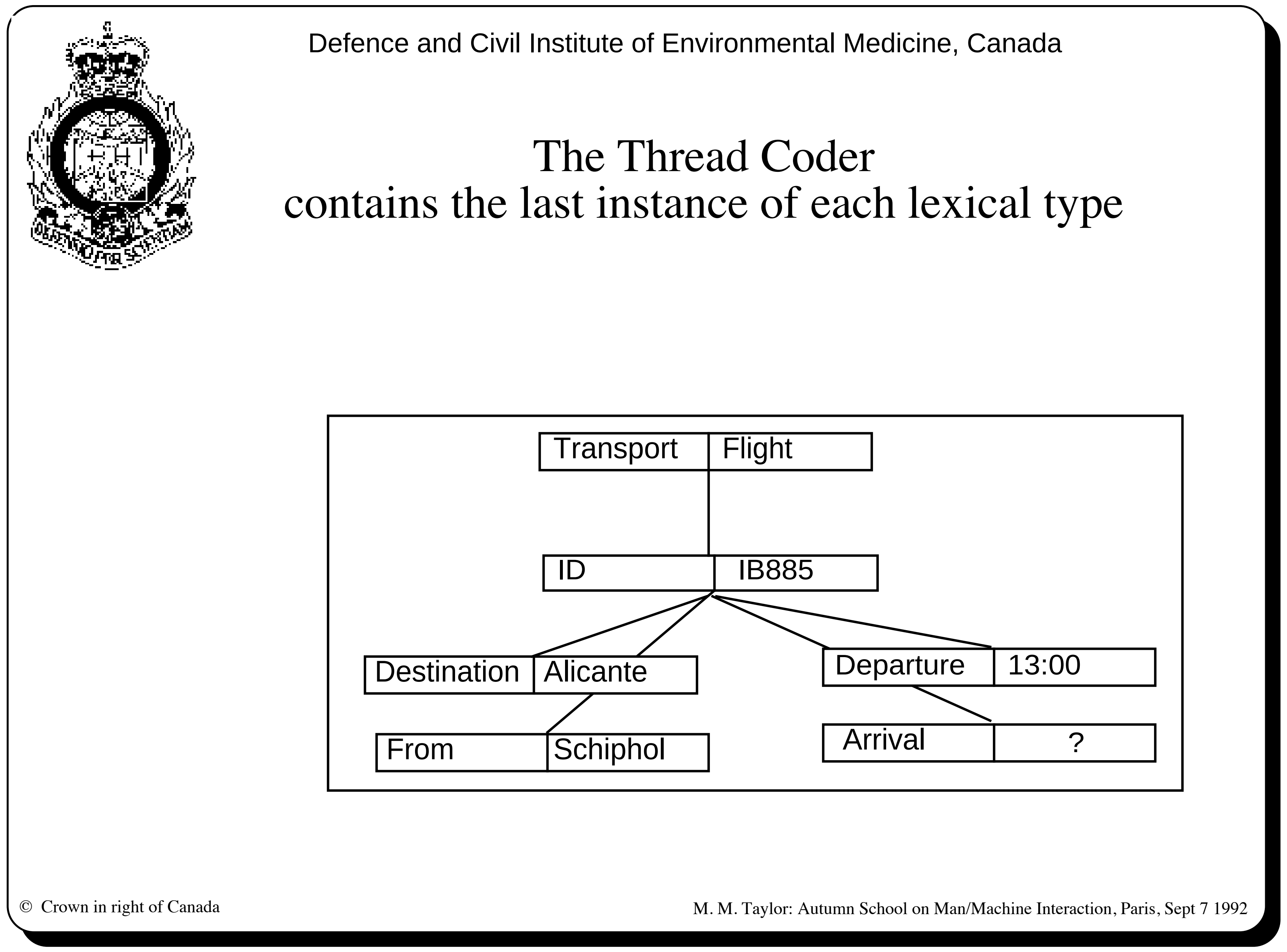
A Problem of Anaphora

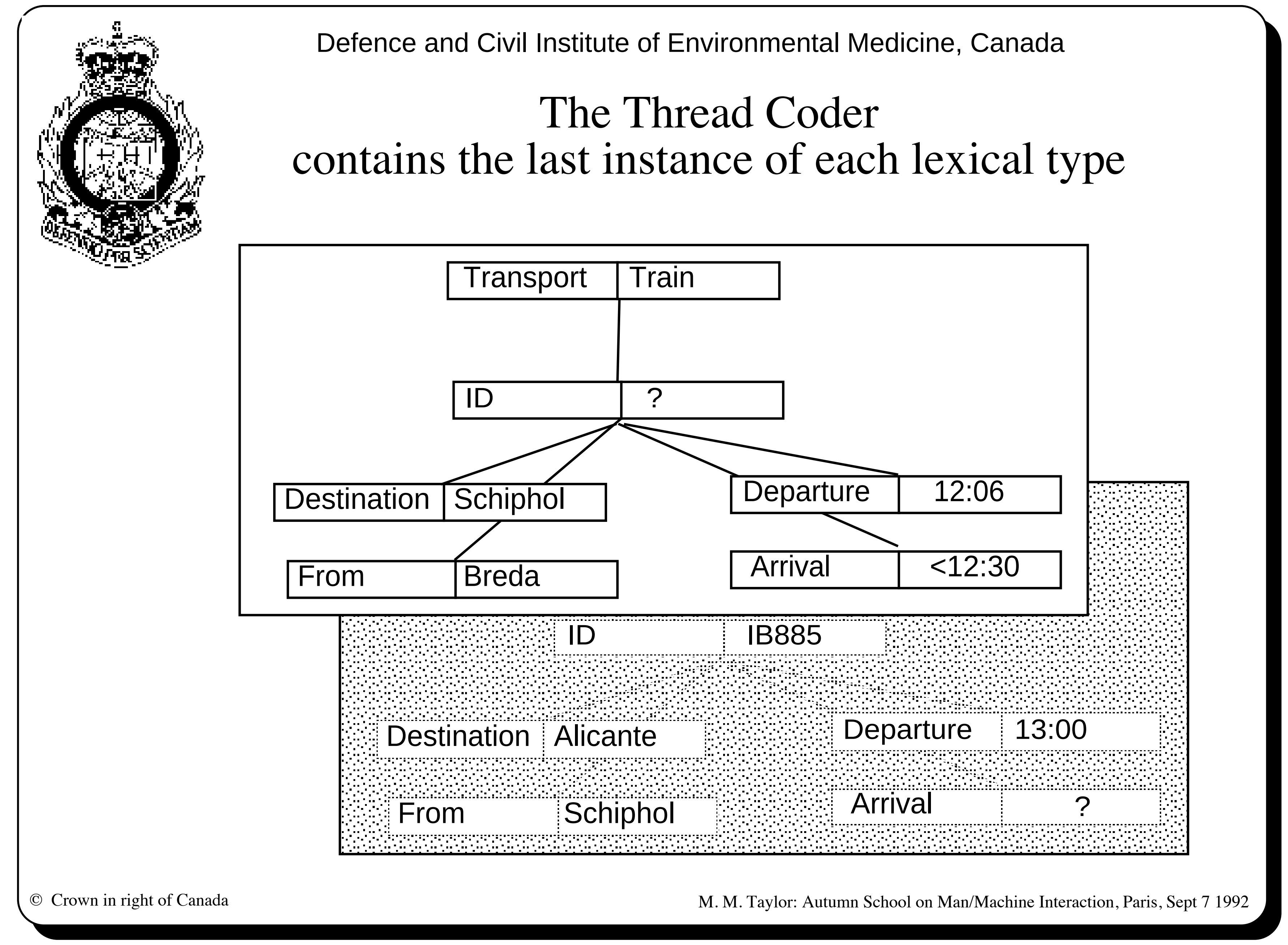
C5: What is the last train from Breda I can take to be in time for flight IB 885?
W6: The train of 12.06.
C6: What is the arrival time in Alicante?
W7: 17.00

Why does the information system respond immediately with the flight arrival time, rather than referencing the train mentioned in the immediately preceding interchange?

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

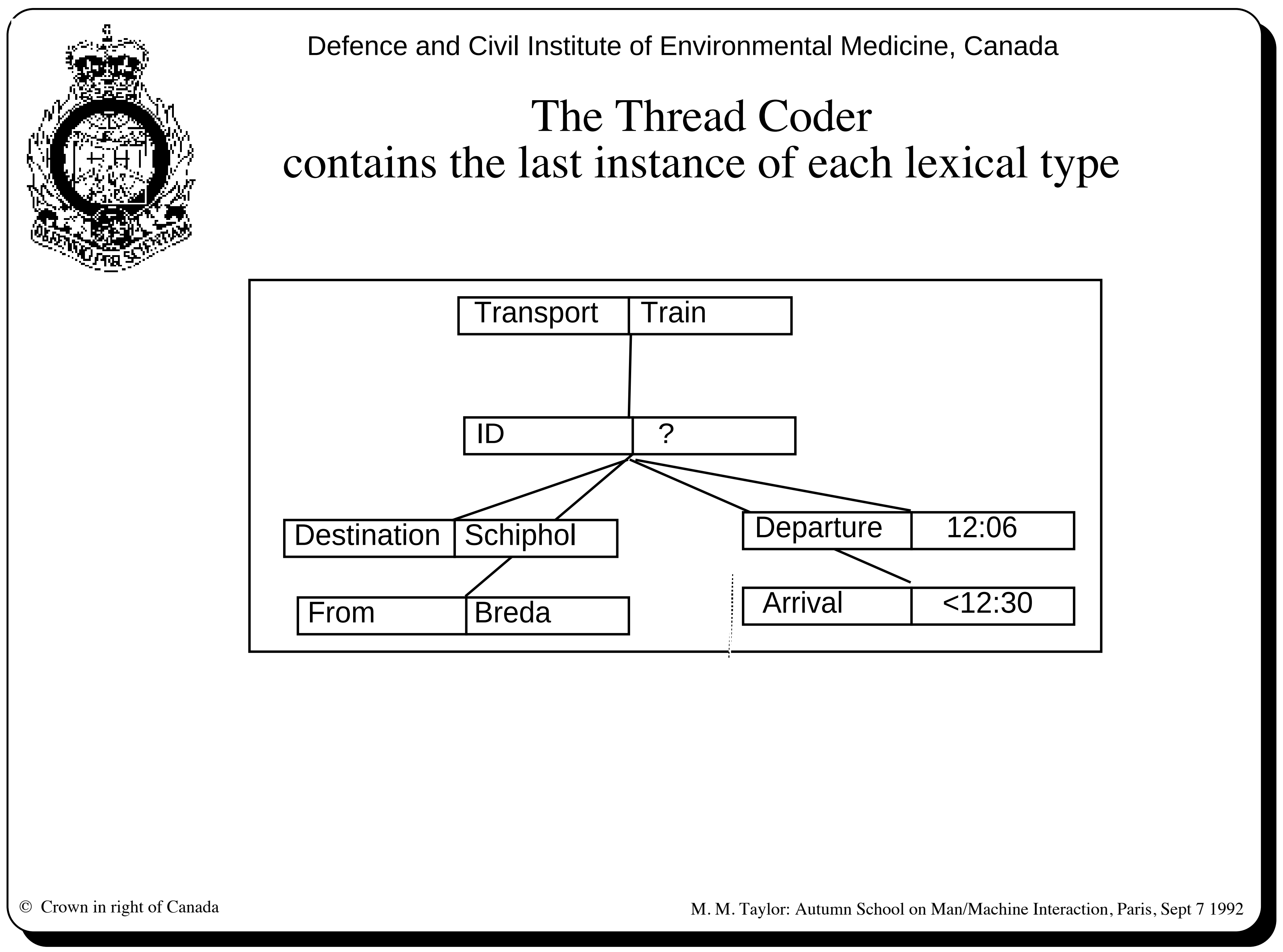
The issue we address here is what it is that allows the Wizard correctly to respond to the question about the arrival time of the plane in Alicante, rather than responding with something like "That train does not go to Alicante." Some analyses have taken it to be necessary for the Information System to be consulted, resulting in a finding that the train does not go to Alicante, followed by a replacement question about the next best possibility, the plane mentioned earlier in the dialogue. In that analysis, possible anaphoric references are stacked, so that if the most recent one does not work, perhaps the next most recent will. We argue that there is a better analysis, that there are two kinds of anaphora, syntactic anaphora, in which the most recently used item of the appropriate lexical type is substituted for an anaphoric or elliptic reference, and pragmatic anaphora, in which appropriate lexical items from the Thread Model that fit the requirements are used, with preference for recently changed items.

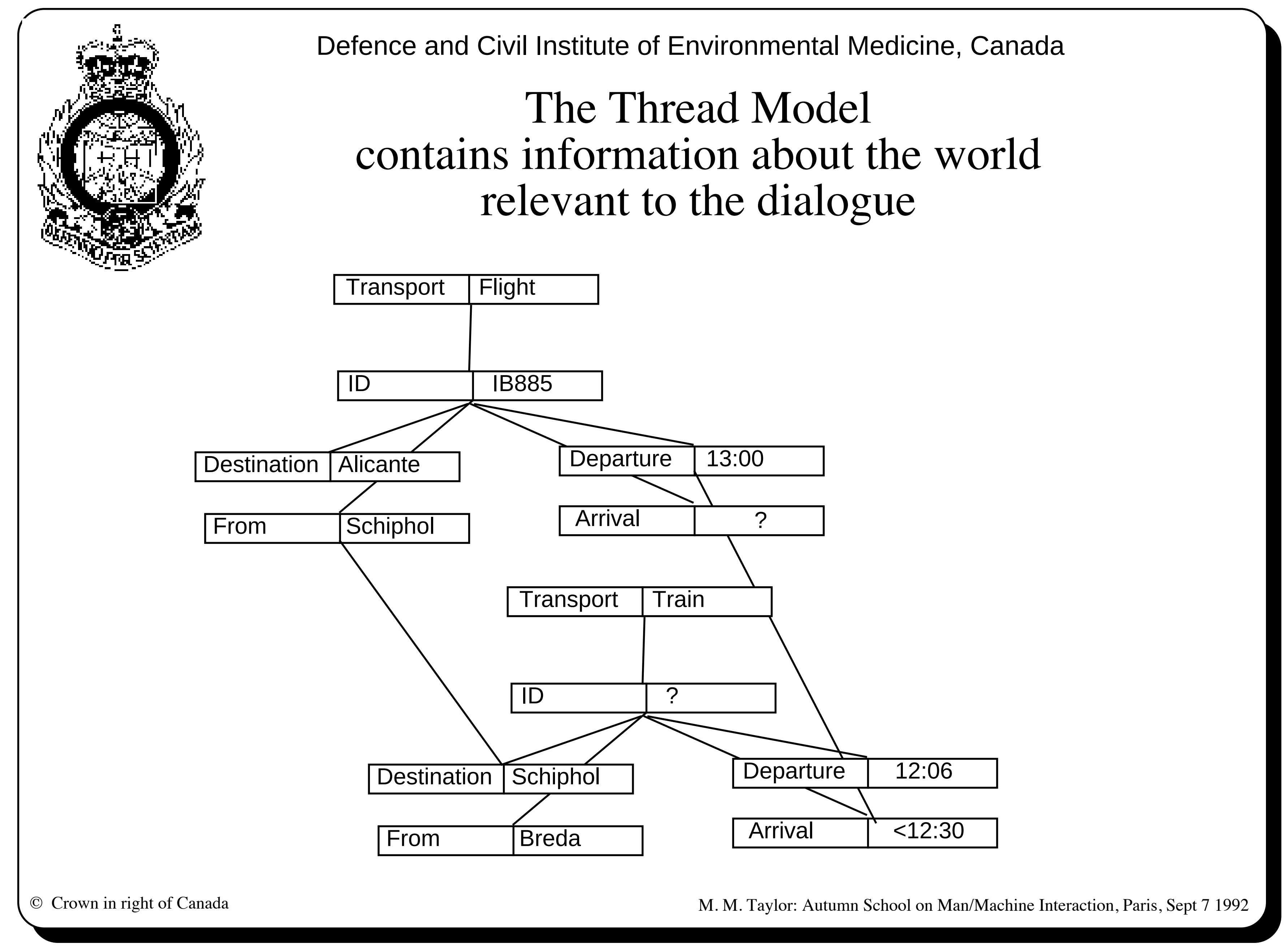




In the Query Assembler protocol, one of the lexical item types might be the transport vehicle. A transport vehicle has several attributes that could be relevant to the query, including its identification label, its origination and destination, and its times of arrival and departure. When the client starts the interaction, the first utterance identifies a vehicle of interest: a plane, ID IB885, departing from Schiphol, departing at 13:00, arriving at Alicante. The arrival time is not specified, but the information provided is enough for the Information System to provide the arrival time if asked. The vehicle is a lexical item in the Query Assembler Thread Coder, and will be automatically referenced if an anaphoric or elliptic reference to a vehicle occurs in the following dialogue.

Later in the dialogue, the characteristics of another transport vehicle are specified: a train, departing Breda at 12:06, arriving Schiphol before 12:30. It overrides in the Thread Coder the lexical item that is flight IB885. The next time there is an anaphoric or elliptic reference to a transport vehicle, the train will be preferentially used to fill out the information omitted in the reference.





Now the Thread Coder contains as its most recent "vehicle" lexical item the train, which is known to arrive in Schiphol. This arrival does not preclude the train also having an arrival at Alicante, since trains pass many stations in their travels, but from the viewpoint of the dialogue, the Schiphol arrival is salient. So when the Client asks the question "What is the arrival time in Alicante?" immediately after asking what time the train leaves from Breda, the question is a little unexpected with respect to the train, though it is not seriously anomalous. If there were no Thread Model, the Query Assembler might well create a query to the Information System requesting the arrival time of the train in Alicante.

In the Thread model, there is much more structure. The train's arrival in Schiphol has been linked inside the dialogue to the plane's departure from Schiphol. The plane is a vehicle with an arrival in Alicante at a time that has not been mentioned in the dialogue, and that therefore may be unknown to the Client. At least the Query Assembler has no information that would mark the arrival time of the plane as being known to the Client, and the Information System has not yet been requested to provide it. The plane has recently been highlighted through its connection with the train, and therefore is a reasonable candidate for anaphoric reference. That it is the correct candidate is probable because of the reference to the unknown time of arrival at Alicante, a pattern that matches the Wizard's model of the Client's model of the vehicle relationships.

Defence and Civil Institute of Environmental Medicine, Canada

Part 11

Multiplexing and Diviplexing

Sharing channels and the rôle of syntax

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

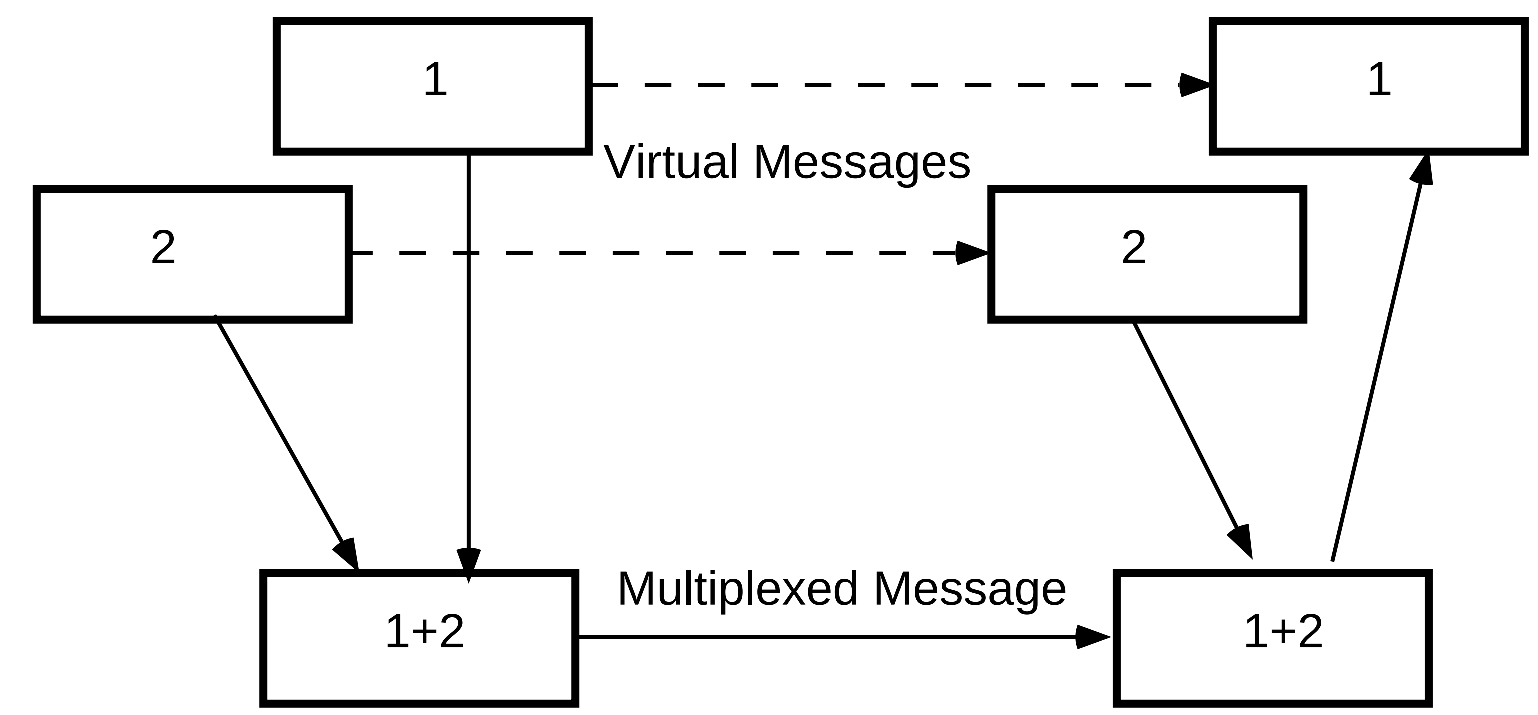
Multiplexing is an engineering term that refers to the combination of two or more messages onto a single support channel. It may mean the transmission of several TV channels on a single cable, for example. In Layered Protocol theory, it refers to the use of a single supporting protocol to handle messages from more than one higher-level channel.

Diviplexing is a word we have coined to represent the opposite of Multiplexing. It refers to the transmission of a single virtual message over two or more separate supporting protocols, such as voice and gesture, or language and pictures. Multimodal dialogue necessarily involves Diviplexing.

This talk is an example of a diviplexed message, using graphic and text supporting protocols. Would it be intelligible if you looked only at the slides, or read the text without looking at the slides?

Defence and Civil Institute of Environmental Medicine, Canada

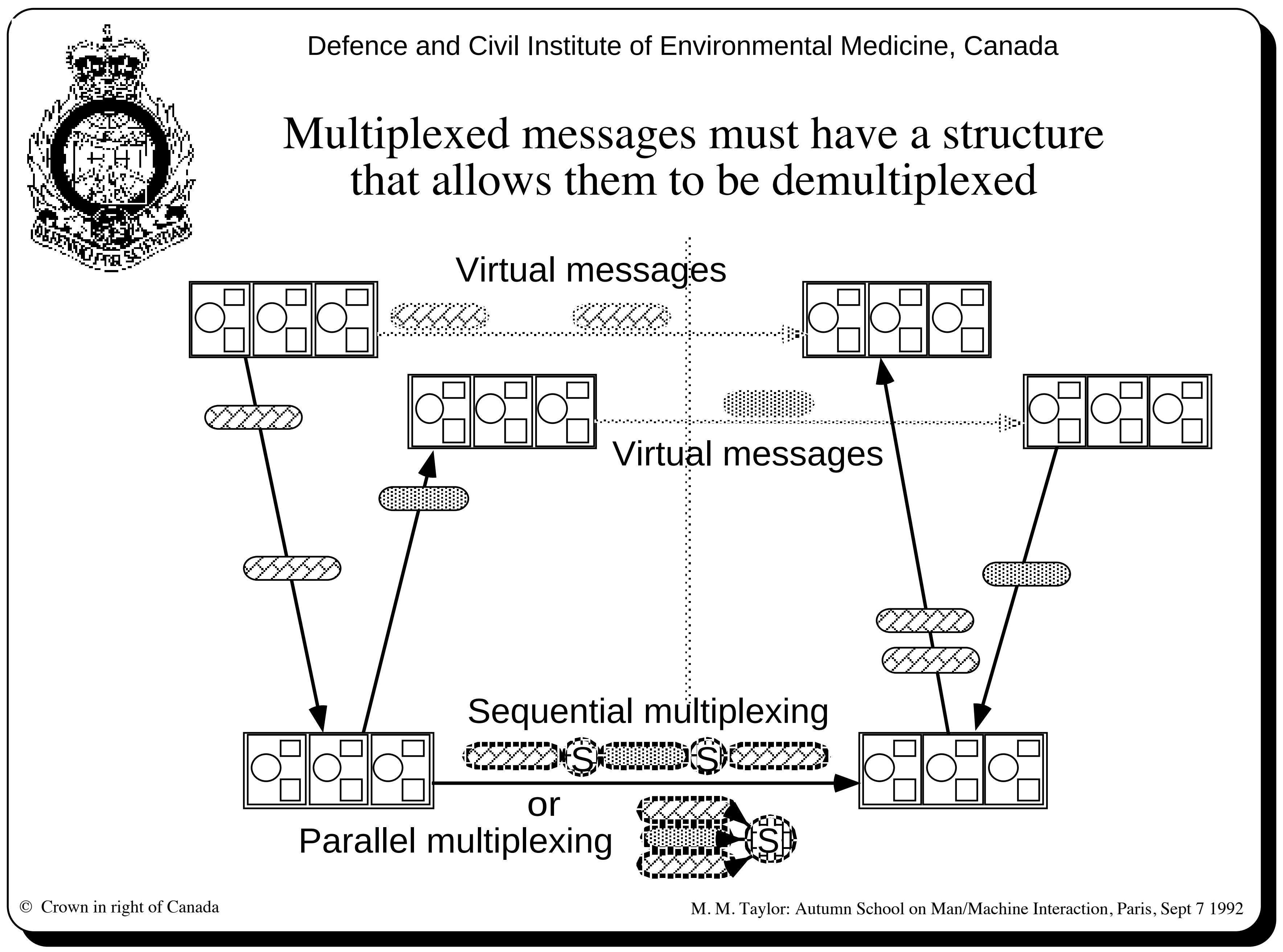
Two or more messages may be **Multiplexed** onto a single supporting channel



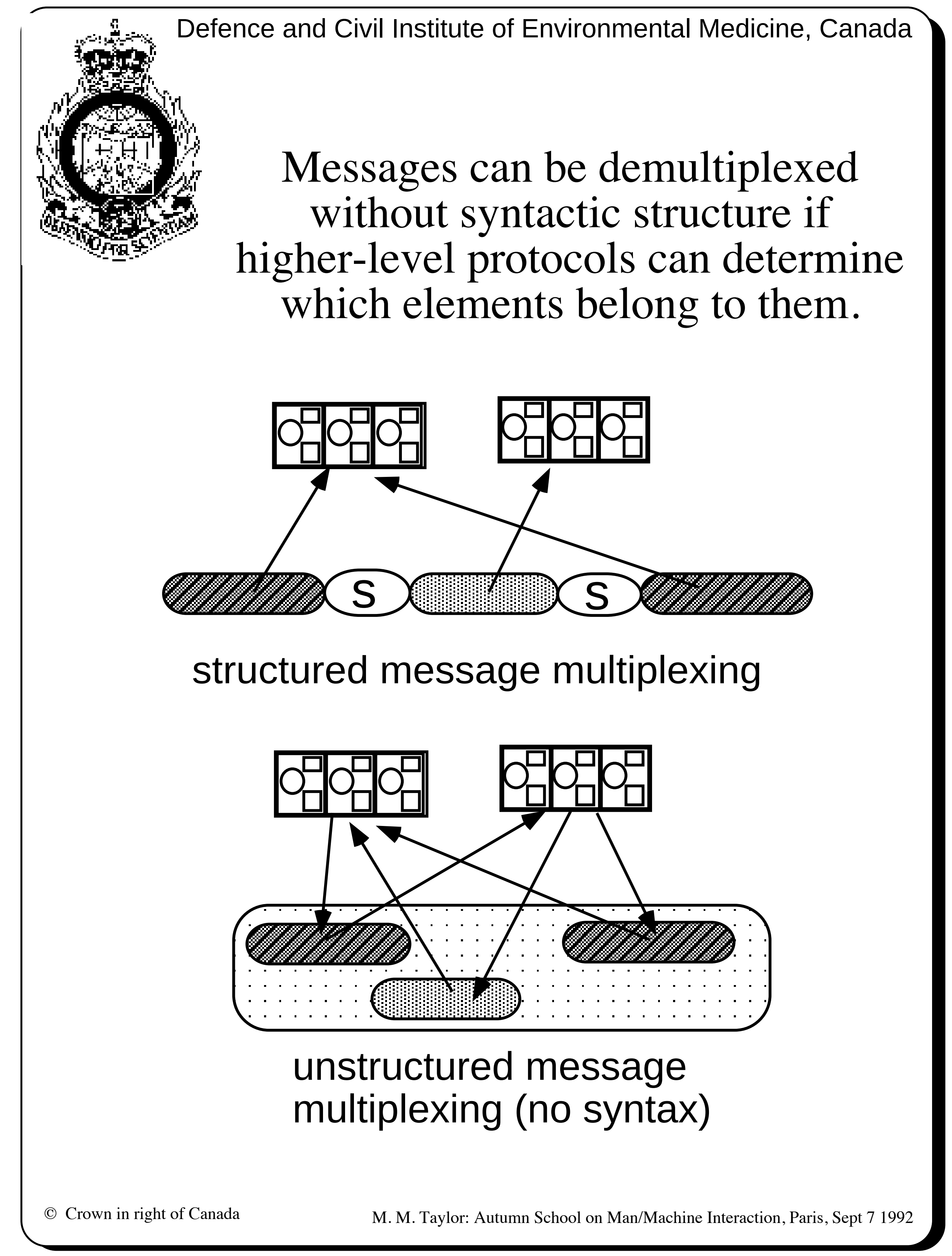
© Crown in right of Canada

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

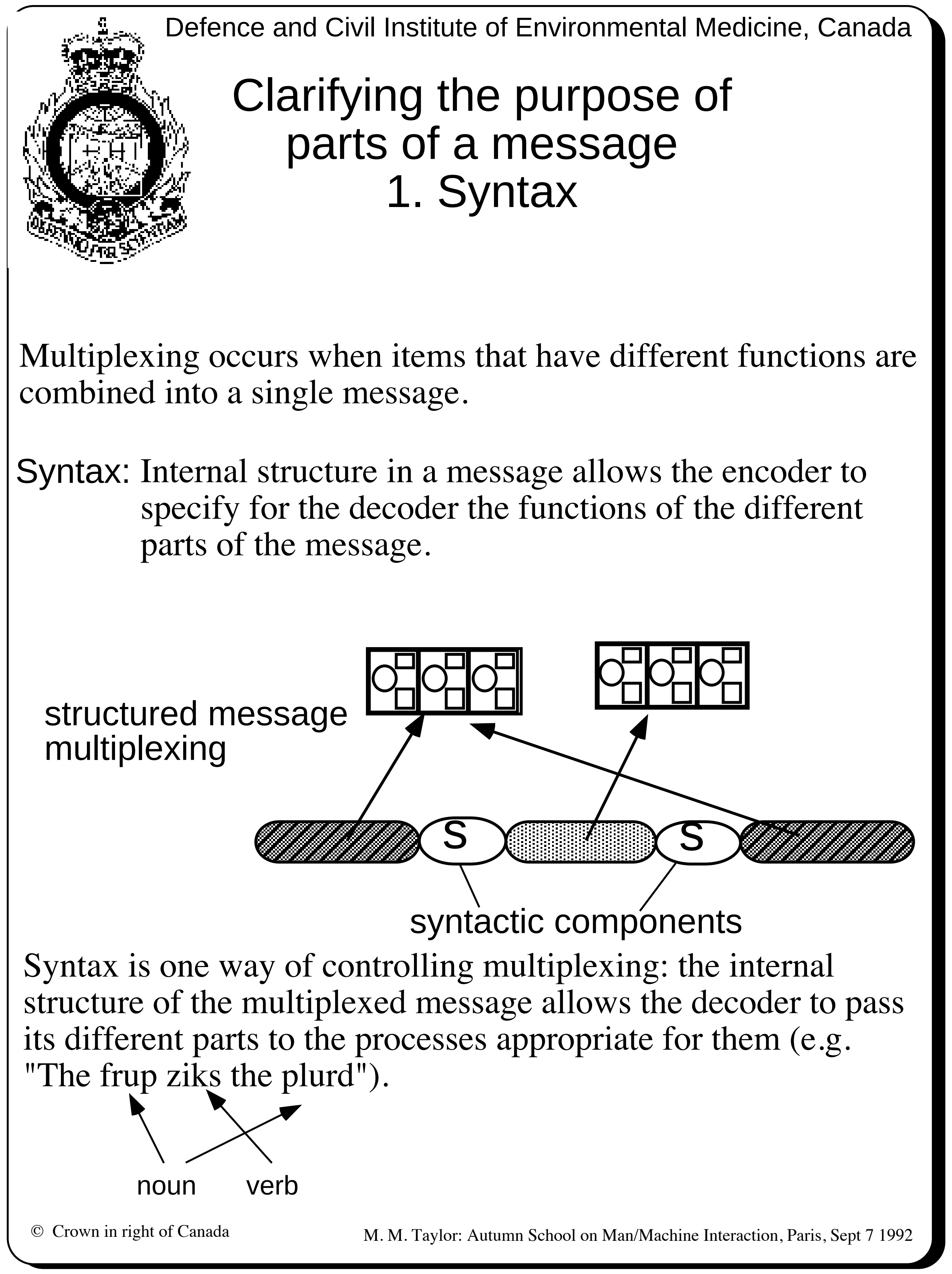
Two virtual messages may be multiplexed for transmission over a single supporting protocol. At the higher level, the two messages are independent, and need carry no information relating to the fact that they are jointly supported. An example might be the interactions with two programs through two windows on a single screen. The only interaction they have is through possible resource limitations in the supporting channel (lack of screen real-estate, in this example). But the lower protocol that supports them must normally contain information (such as the window frame elements) that allows the two virtual messages to be demultiplexed at the recipient's end. This information could be that the kinds of lexical elements used for the two messages are quite different, but usually the supporting protocol will structure the combination in some way that allows it to redistribute the messages to the proper higher protocol regardless of the actual content.

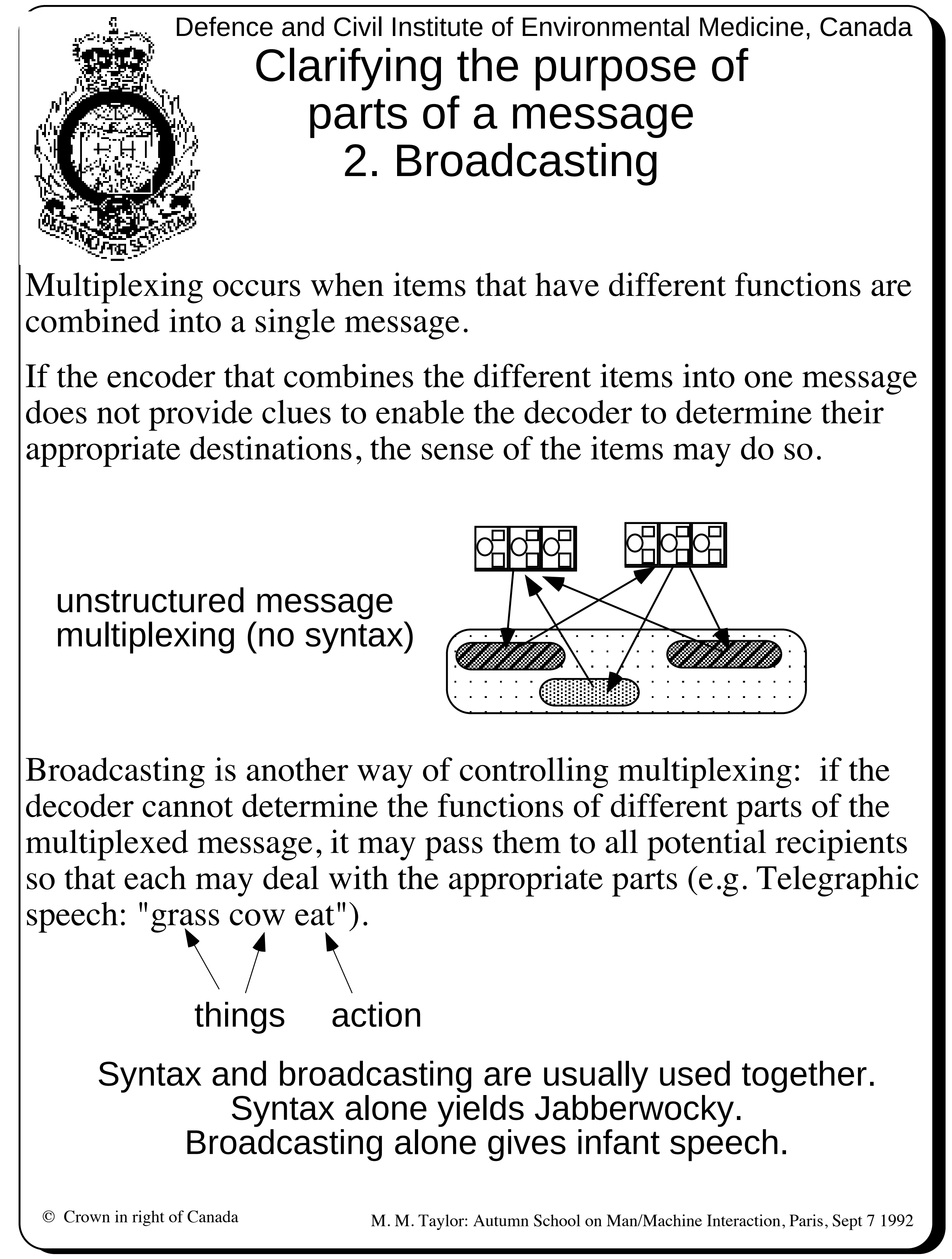


Multiplexed messages may be sent either in sequence or in parallel, or in a combination of the two. Ordinarily the combined message will contain some added message elements that describe which parts of it belong to which higher-level message. The function of these added elements is part of the protocol, accepted by both parties, perhaps as a general convention. It is therefore an aspect of the redundancy of the multiplexed combination, and an aspect of the syntax of the lower protocol. But more than this, it indicates the relationship of the "content" parts of the combined message and therefore performs what we ordinarily think of as a major function of syntax. Syntax therefore is seen as having two distinct functions. Firstly, it provides the redundancy that allows for many errors to be corrected at the level at which they occur, and secondly, it allows the functions of the different components of a message to be determined without regard to their actual contents.



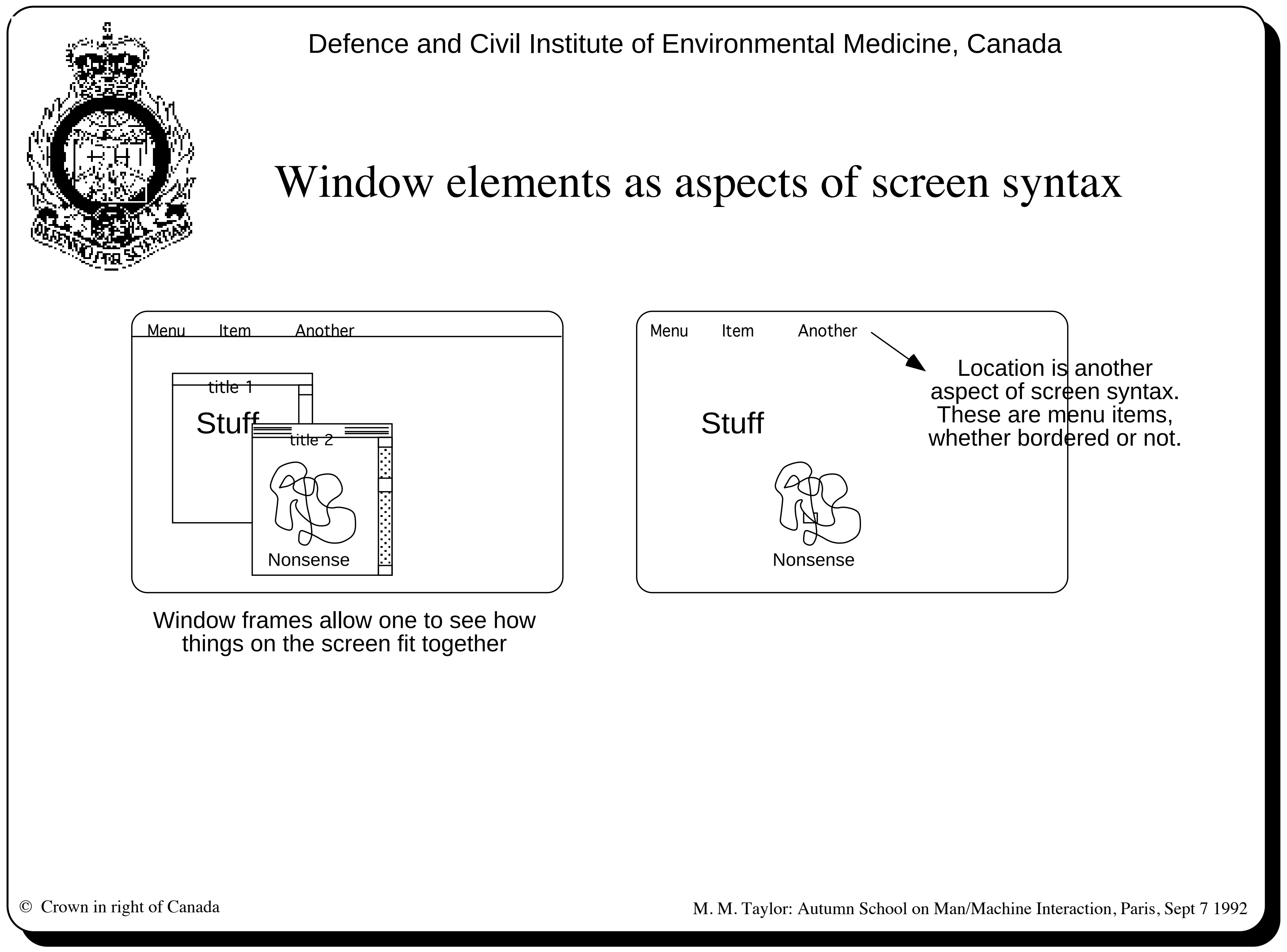
Syntactic structure is not always required for demultiplexing. It can happen that the different components of the combined message are sufficiently distinctive that each of the higher-level messages can accept only the components that belong to them. In such a case, the lower-level protocol need not use syntax, but can broadcast each of the elements of the combined message to all the protocols it supports. If the elements of the different high-level messages are not sufficiently distinctive, broadcasting may lead to difficulties of interpretation in the higher protocols.





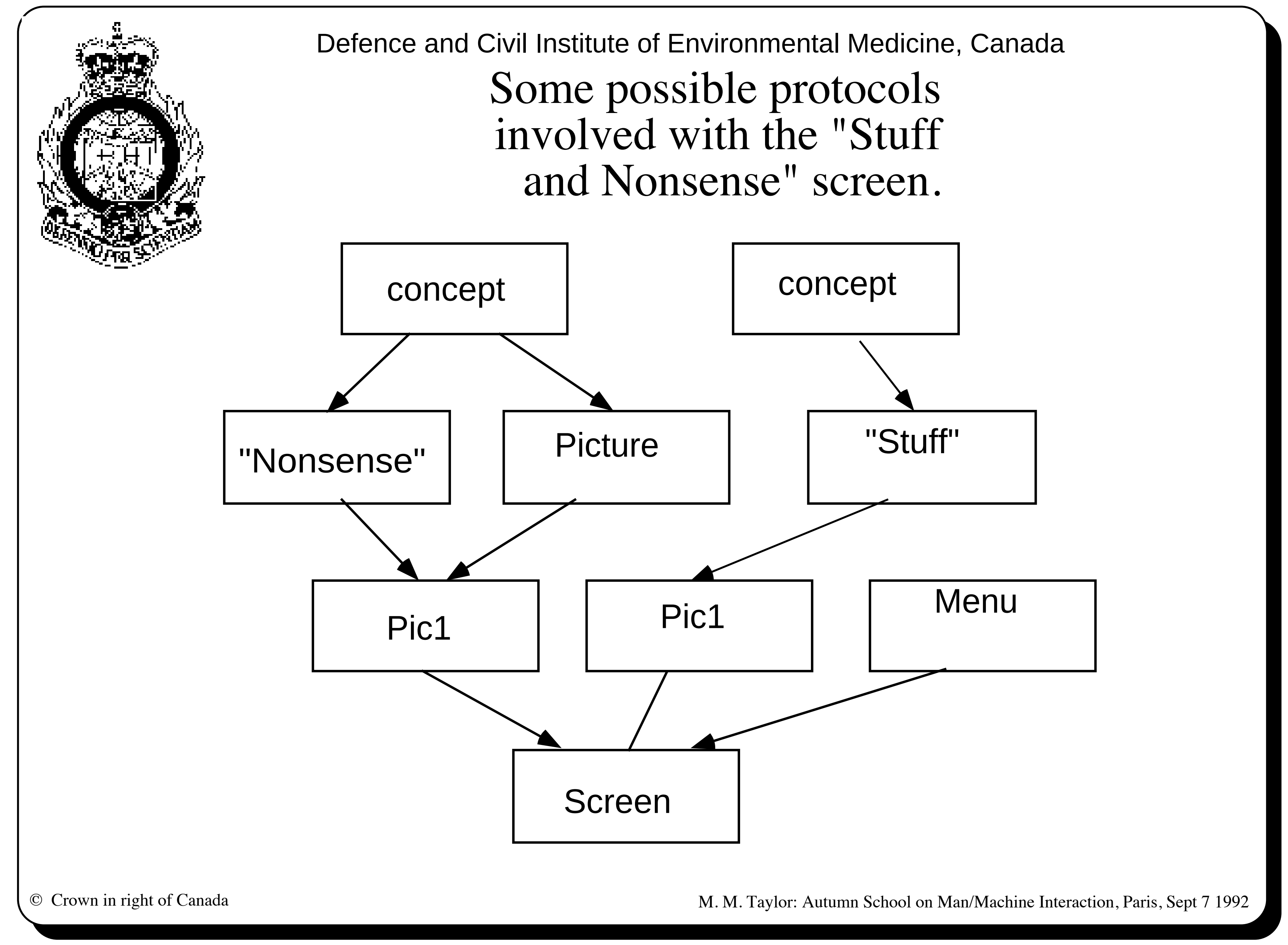
It is possible to relate the usual sentence-grammar use of *syntax* to the general case: the information that allows the functions of different parts of a message to be determined independent of their content. Consider the kind of language typified by Jabberwocky—"T'was brillig and the slithy toves did gyre and gimbal in the wabe"—in which it is clear that "brillig" refers to a situational variable, such as an aspect of the weather, or possibly time, that "toves" are active, probably animal, and "slithy" is an adjective describing them, etc. The actual content is not clear, but the relationships among the content words are fairly well defined, as are the relationships among the perceptions they evoke.

The opposite kind of language is sometimes called "telegraphese." Children use it in their earliest utterances of more than two words. The salient content is mentioned, and the situational context must be used to determine the function of the words. Consider "Grass cow eat" which could be a child's description of a farm scene. The three words would fit together as a cow eating grass, but could refer to a donkey eating a statue of a cow made of grass. It could even refer to some criminals not eating, being cowed by fear that a colleague had informed ("grassed') on them. The functions of words in telegraphese must be discovered from their sense and the pragmatic situation, not from their syntactic functions, which are not specified with any precision.

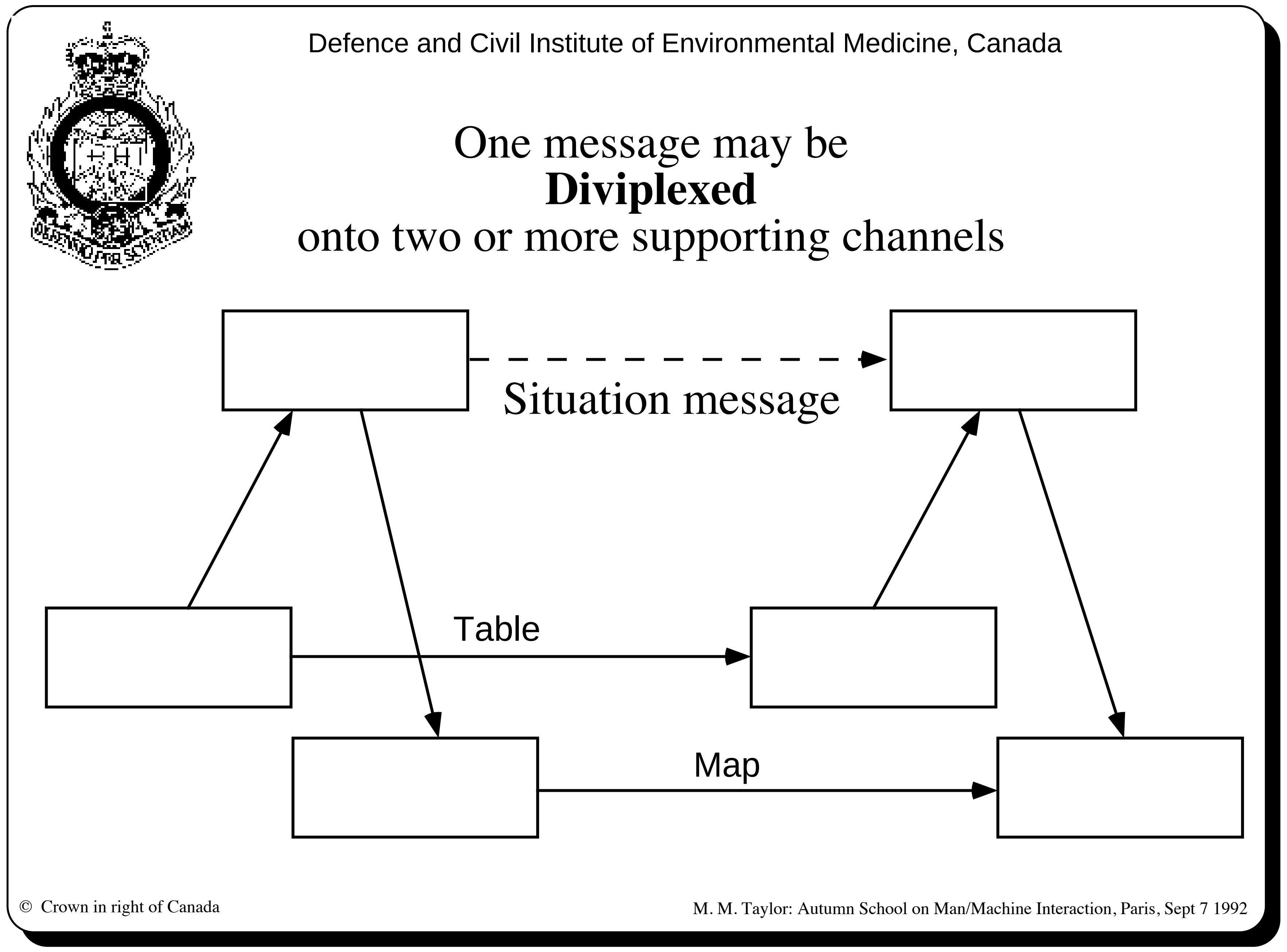


The popular Window interface provides an example of syntax for multiplexing and demultiplexing. The messages from two sources are shown on a single screen. The window frames show which components belong together. The earlier alphanumeric displays showed only lines of print, and when a time-sharing computer produced an output it was often difficult for the user to determine which of several running programs was responsible. Windows provide added information, unrelated to the program, that allows the user to separate out the streams that come from different programs.

In addition to the window frames, location on the screen sometimes provides syntactic information within the screen protocol (as does word order within English sentences). The Menu items at the top of a Macintosh screen provide an example. The positional aspect of screen syntax also was sometimes used in the provision of a control line at the bottom of some alphanumeric displays.

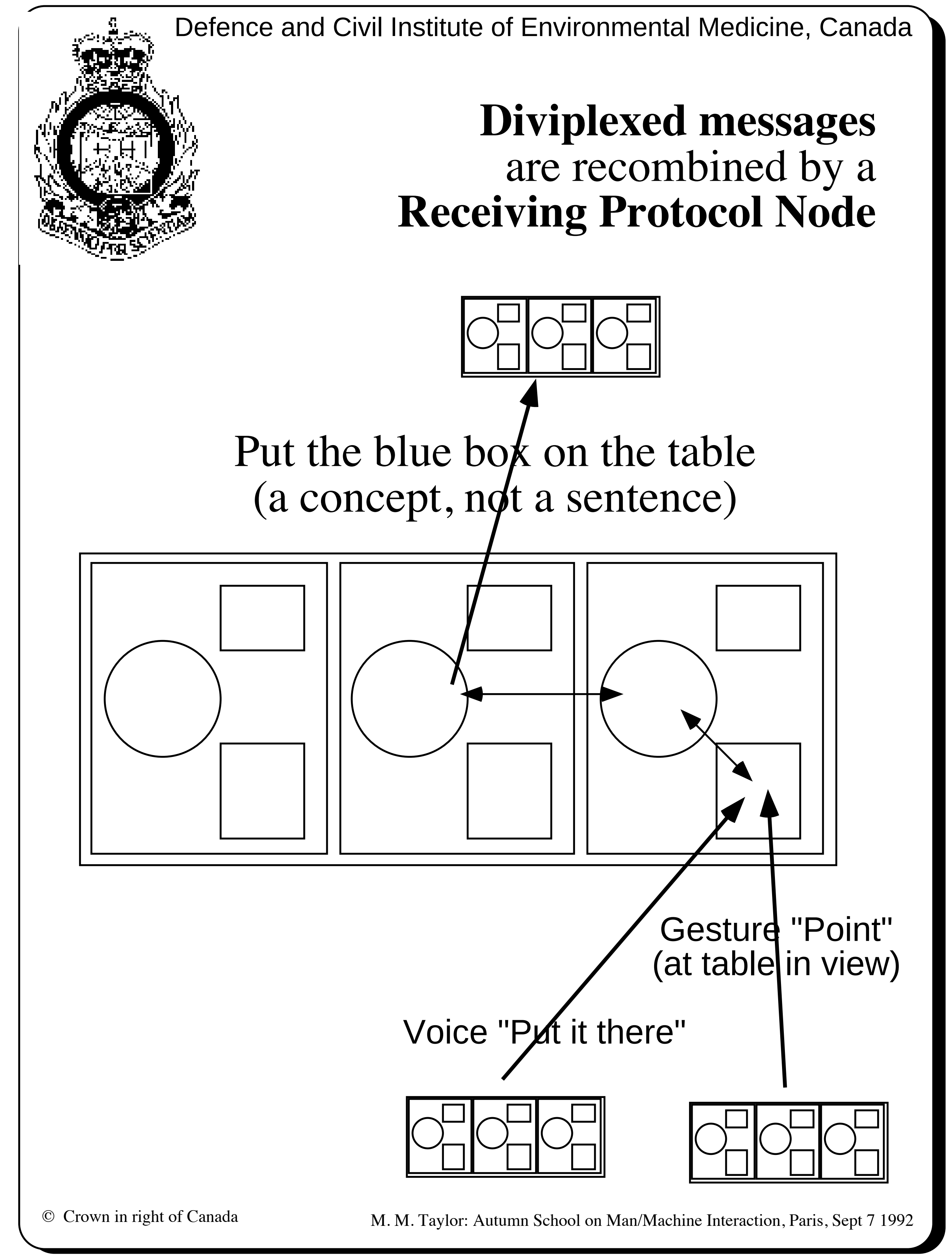


The apparently simple structure of the previous figure depends on a fairly complex web of protocols. Those shown here are a minimum. There are at least two separate concepts (presumably processes) involved, one of which provides the verbal and pictorial output shown as "Nonsense" and a scribble in the last figure. The other provides the linguistic output shown as "Stuff." The "Stuff," and the "Nonsense" together with the scribble, are converted into pictures, and the menu forms a third picture. Finally, all three pictures are placed on the screen, along with syntactic pictorial elements that belong to none of them, but that show which parts of the screen display go together and what functions they have.



Diviplexing is the opposite of multiplexing. In this example, a message describing a situation is sent partly as a table of data and partly as a map. The Recipient receives and interprets the map, and independently receives and interprets the table of data Recognizing, perhaps from information internal to each of the supporting messages, that they form part of the same higher-level situation message, the higher-level protocol node combines them to interpret the situation message.

The situation message is not complete if either the map or the data table is interpreted alone. Each has something missing, and may well include pointers to information held in the other. The table may have a column of coordinates, for example. Such reference pointers are analogous to anaphora in a single-channel message; information omitted in one channel but supplied in the other is akin to ellipsis.



In the next two figures we use the classic example of a multimodal utterance: "Put that there" to draw the analogy between diviplexing and anaphora or ellipsis. We deal with a situation in which "that" refers to a specific blue box, and "there" is a table top. We will show how, depending on the talker's beliefs about the listener's beliefs, as well as on the pragmatic situation, different unimodal but anaphoric or elliptic phrases may be used, or equivalent multimodal utterances may be more appropriate. Information provided by a non-linguistic channel may perform exactly the same function as information believed by the talker to be in focus in the listener's memory .

Defence and Civil Institute of Environmental Medicine, Canada

Put the blue box on the table
(some alternate forms)

1. Purely linguistic forms

Speaker's beliefs

The blue one
Hearer knows a box is to be placed, and where it is to be placed, but does not know which box it is.
Ellipsis (box, put, table)

On the table
Hearer knows what is to be placed, and what to do with it, but not where to put it.
Ellipsis (blue, box, put)

Put it on the table
Hearer knows what is being handled, but not what to do with it and where.
Anaphora (blue box)

Put it down
Hearer knows what is being handled, but not what to do with it.
Anaphora (box) Ellipsis (table)

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Defence and Civil Institute of Environmental Medicine, Canada

"Put the blue box on the table"
(some alternate forms)

2. Diviplexed forms (language and gesture)

Gestures can be used in place of or in conjunction with anaphora or ellipsis

There *Point*
Speaker and hearer know that the blue box is to be put somewhere. (Hearer may be carrying it).
Ellipsis (put, box)

Put the box over there *Point*
There is only a blue box, but there are some other objects that might be handled.

Put that there *Point Point*
The "classic" mix of pointing and speaking. Both referents are disambiguated by gestures. (If the first "Point" were to be omitted, "that" would be an anaphoric reference to the box.

Put it there *Point*
Hearer is holding the blue box, and talker must indicate what to do with it. Only the single ambiguous referent need be disambiguated.
Anaphora (box)

© Crown in right of Canada
M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

These figures provide eight examples of purely linguistic and linguistic-gestural ways of indicating that the listener should put a particular blue box down on the table.

The first four examples are purely linguistic, the different forms relying on the talker's beliefs about what the hearer has in focus and what the hearer does not know. The second set of four examples also relies on the talker's beliefs about the hearer's knowledge and focus, but now the talker supplies some of the information omitted from the utterance by gesturing, rather than expecting the listener to supply it from memory.

The analogy between diviplexing on the one hand and anaphora and ellipsis on the other should be clear from these examples. The same kind of effects occur at all levels of abstraction, not just words, though they are easier to detect at high levels than at low. The benefits for speech understanding of seeing the face of the talker in high-noise environments illustrate diviplexing at a very low level. The largely unconsciously made hand gestures that accompany much conversation may have a similar use. At a high level, some elements of a largely verbal argument may be accomplished better by visual demonstration than by verbal description.

Defence and Civil Institute of Environmental Medicine, Canada

Part 12

Final Thoughts

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

To conclude the talk, we recapitulate, and provide a few additional final thoughts for consideration.

Firstly, it is important remember the two mottos: *All Behaviour is the Control of Perception*, and *All Communication is the Control of Belief*. If there is ever a problem in the design of an interface or the analysis of a dialogue, think: What perception(s) or belief(s) is the person trying to control? If we analyze problems with interfaces, they usually come down to a failure to provide the user with the information that would allow the control of an important perception. The user is required to use faulty memory, or is unable to predict the action of the computer, or some such.

In considering the twin mottos, the ubiquity of layering in human behaviour or communication must be kept in mind. Control is continuous, not discrete. It is rare that one has to await the results of one action before commencing the next. More commonly, the effects of behaviour on perception are continuously monitored at many levels of abstraction or complexity, and the lower-level references (goals) are altered continuously to reflect the changing higher-level error signals. Interfaces must allow for interruption and change of activity as the demands of the higher-levels change.

The hierarchic structure of perception or belief leads to some claims about learning that were not touched on during the main body of the talk. In particular, PCT proposes exploration (typically under the rubric of "reorganization') as an effective way of learning. An interface should support exploration—learning by doing, rather than requiring the user to discover from a manual what to do to get useful results. But there is a problem with learning by doing, as many moderately skilled users know. Once a successful method of doing something is learned, exploration may stop, and more effective ways are not discovered. The good is the enemy of the best, even when there would be no inherent difficulty in achieving the best. Book learning can and should supplement learning by doing, but it should not substitute for it. Manuals should supplement, not replace, explorable interfaces, and the explorable interfaces themselves should guide users toward potentially interesting ways of interacting.

A Layered Protocol design can facilitate learning by providing easily learned and consistent low-level protocols to support many different kinds of high-level ones. This was one of the basic ideas of the Macintosh; the Toolbox was provided to developers in the hope that they would use consistent low-level interactions no matter what their applications. LP goes further, in that more complex protocols can be substituted for simple ones at any point in the hierarchy without disturbing the rest, thus allowing the learning of a complex interface to be made modular.

12-1

Defence and Civil Institute of Environmental Medicine, Canada

Design vs. Rapid Prototyping

Design:

Assumption that user's reactions are predictable. Long delays between constructor's acts and feedback as to the effects of the acts.

Rapid Prototyping:

Assumption that evolutionary changes will lead to an optimal solution. Rapid feedback as to the effects of constructor's acts, but no assurance that evolution does not lead to dead-end local optimum.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Perceptual Control Theory was developed to explain how a living organism could survive in an unpredictable world. Only by continually monitoring whether identifiable complexes in the world remained within tolerable bounds, and by acting to ensure that they did, could the organism compensate for disturbances that would destroy a non-living thing of equal fragility.

12-2

Defence and Civil Institute of Environmental Medicine, Canada

Planning vs. on-line Control

Planning:

Assumption that the effects of actions are predictable. Long delays between planning decisions and feedback as to the effects of the decisions.

On-line Control:

Assumption that smooth changes will lead to an optimal solution. Rapid feedback as to the effects of most acts, but no assurance that smooth change does not lead to dead-end local optimum. Requires additional sensory information.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Layered Protocol Theory was developed to explain how communication could be conducted between partners unable to be sure of the effect either's actions would have on the other. Only by monitoring the effects of complex messages could each be assured that the other was coming to believe what they were intended to believe.

The world is not totally unpredictable, and neither are communicative partners. The world is more chaotic than random. To a certain extent, pre-planned actions can have the desired effect with high probability. Preplanning is essential if ill-considered actions could lead to disastrous consequences. Likewise, it is reasonable to design interfaces using reasonable assumptions about the behaviour of certain classes of future users, with some assurance that the performance of the interface will not be disastrous.

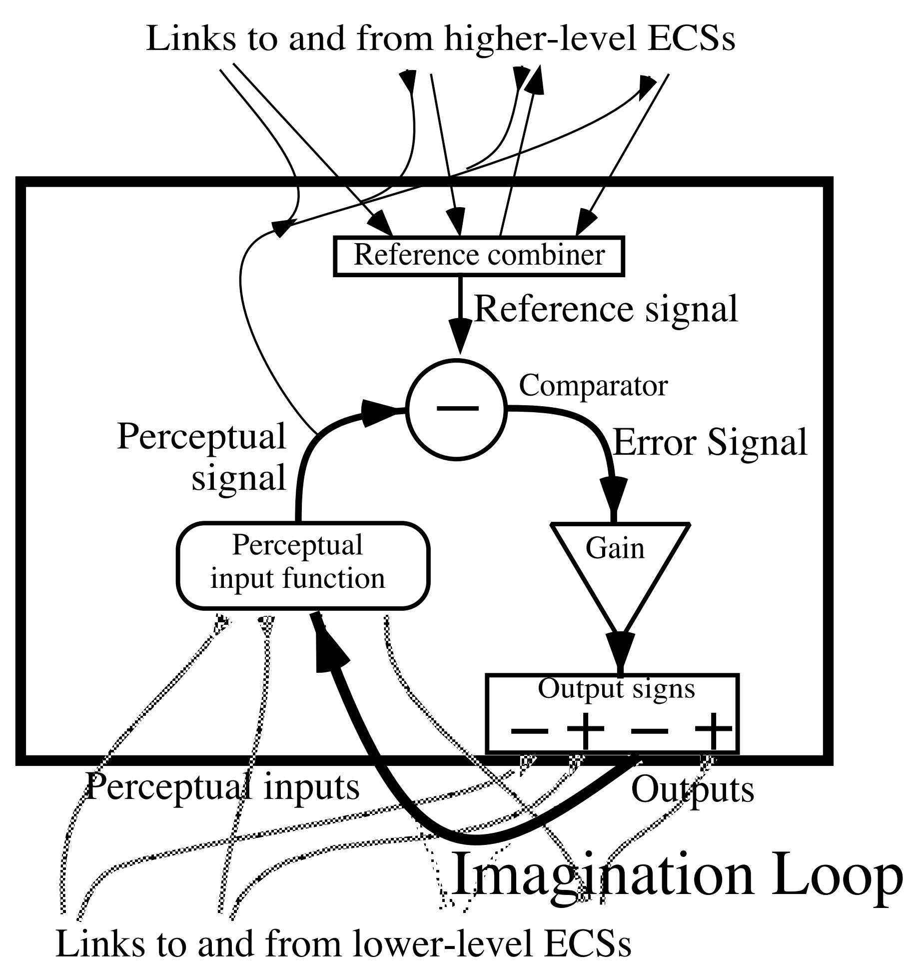
The best laid plans of mice and men gang aft agley, as do the best designs of mice and screens. For these situations, only on-line control or rapid prototyping are effective. They will usually work eventually in any case, but the evolutionary development of an effective control network or interface design for specific situations can run into a dead end if it starts from a ill considered base. And sometimes the consequences of a mistake in the evolutionary development of a plan or a prototype could be fatal. As with exploration, the good can be the enemy of the best, unnecessarily.

12-3

Defence and Civil Institute of Environmental Medicine, Canada

Planning and PCT

Planning uses the "imagination loop" of an ECS. It determines "what if" the desired percept could be achieved at a lower level. On-line control uses the world as its test-bed.



M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

The way planning is done within PCT is through the imagination loop. The results of successfully completing certain actions are evaluated, and can be done much faster than those actions would happen in the real world. There are no real-world dynamics to constrain imagination. But then neither are there any real-world conflicts to make actions incompatible. One can imagine eating one's cake and having it, too. So plans sometimes fail when applied to the real world, even in the absence of unforseen disturbances. There is no substitute for real-world validation of plans and designs. But when failure would be disastrous, there is no substitute for careful planning, either.

12-4

Defence and Civil Institute of Environmental Medicine, Canada

Features of Layered Protocols

No "Dialogue Control Module." Control distributed through many protocols that have only local concerns.

Distributed representation of syntax, semantics, and pragmatics, which are cleanly distinguished. They are not layered, but are aspects of every protocol.

Global and local coherence are distinguished for each protocol. What is Global Coherence for one protocol is Local Coherence and syntax for another.

Anaphora and ellipsis are integral components of each protocol, related to belief structures.

Partner models and dynamic belief structures are separated from coding mechanisms.

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

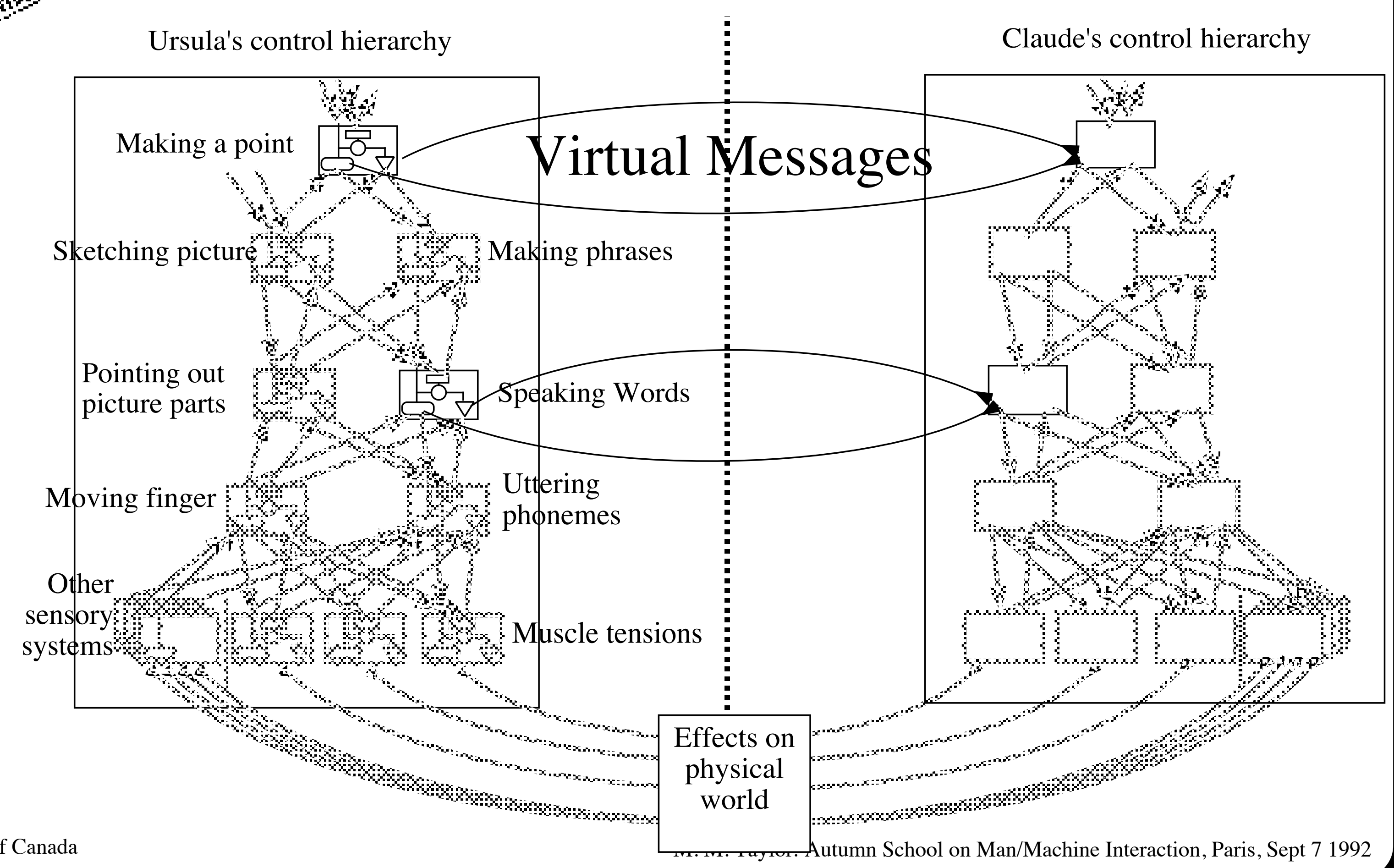
Layered protocol designs differ from most conventional interface structures in providing no place for a construct that could be called a "dialogue control module." They differ from most layered analyses of interaction by not segregating lexicon, syntacs, semantics, and pragmatics into different layers. They provide clean descriptions of global and local coherence, and relate anaphora and ellipsis to the roles of different channels in a multimodal interaction. Finally, there is no place for fixed encodings of elements from layer to layer, though conventional encodings (syntax) are accommodated as normal constructs.

12-6

Defence and Civil Institute of Environmental Medicine, Canada

A Layered Protocol hierarchy

(Ursula and Claude can be read equally well as User and Computer)



M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

Defence and Civil Institute of Environmental Medicine, Canada

Things to remember

Feedback implies recursive partner modelling

Feedback implies Layered Protocols, if real-time interaction is required

When thinking about what someone is doing at a computer interface, think more about what they are wanting to perceive.

When designing or producing a message, think about what the originator wants to believe about the partner

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

12-5

Defence and Civil Institute of Environmental Medicine, Canada

Basic Principles

Psychology

All behaviour is the control of perception

Layered Protocols

All communication is the control of belief

M. M. Taylor: Autumn School on Man/Machine Interaction, Paris, Sept 7 1992

These final three figures represent the main ideas that the talk is intended to get across: the two mottos, the idea of layered hierarchies interacting by means of virtual messages carried through a physical world, and some consequences.

THE END