FORMAL METHODS IN SYSTEM ANALYSIS

A.W. Holt

Rapporteurs: Mr. L. Marshall
Dr. S.F. Ross
Dr. J. Scott

Lecture 1

The speaker's areas of interest are defined and briefly discussed, beginning with an introduction to the concept of Petri-nets, and their application to system analysis.

In this talk I will discuss some aspects of system analysis, in the context of the overall title of this Seminar. I will ask about the nature and relevance of systems, and what facts about systems should be known to students. I will present a view of the system subject, and give some high priority items about which the general citizen should be aware. The following five points summarise the scope of my talks:

1. What is the system subject, and why is it important?
2. The principal concepts of the system subject.
3. What point of view on computing and information does it suggest?
4. What should the average citizen not fail to learn about computing.
5. Special topics.

In pursuit of 1 above, I shall now read two short papers. The first I gave last summer at the M.I.T. Conference on Petri-nets and Related Methods; the second is a set of remarks about Computer Science which I was provoked into making some years ago at a technical meeting.

Petri-nets and Systems Analysis

The title of this conference is "Petri-nets and Related Methods". I would like to address the question: methods for what? In the first part of my talk, I would like to contribute to this conference my vision of the applied field to which our efforts are - or should be - addressed. In the second part, I would like to show in what way that vision is especially related to Petri-nets.

As many of you know, I have been occupied with Petri-nets for about eight years, and engaged in research with the same general intention for about ten years.

That general intention was - and is - to develop concepts and notations for the study of "systems". However unsure I was at the start about what, in a technical/theoretical sense a system might be, I was quite sure that the applied sense of the term should somehow cover computer and programming systems as I know them professionally - at least cover many of their significant aspects. Should it also cover significant aspects of economic systems, governmental systems, traffic systems, ecological systems, social systems, etc.? I had very little idea. What I did believe was that my choice of focus - its direction, breadth and depth - could easily spell the difference between barren and fruitful work for many years to come.

Six-or-so years after I had started, I felt I began to apprehend the applied subject toward which the theory building efforts were groping. Systems, I thought, are to be understood as an aspect of human organisations - more exactly the humdrum, mechanical aspect of them, the aspect expressed in organisational rules that lay down duties, rights, discretions, physical and procedural forms, routes, routings and routines, schedules, storage and security arrangements, access permissions and denials, accounting disciplines - fiscal, and other - etc. Other aspects of organisations, such as the prevailing attitudes of its functionaries and their job morale, over all organisational traits such as efficiency, stability, adaptability, dependability, productivity, creativity, growth curve, etc., lie outside of this focus, though no doubt affected by the mechanical aspects to which I am drawing attention.

What most meets the mind's eye in looking at organisational mechanisms is that they all concern the relations between a multiplicity of people and things in their organisational capacities -

relations that express themselves in terms of expected patterns of interaction; one may also say: dynamic relations between a multiplicity of parts.

Now notice that all the examples above of the mechanical aspects of organisation are relevant to the description of the interface between one organisation and another, relevant, in other words, to the description of the relations between a multiplicity of organisations. These remarks naturally suggest hierarchic constructions.

The inner mechanical workings of an organisation are descriptive of its constitution of sub-organisations and their relations to one another. The outer mechanical working of an organisation is descriptive of it - taken as a unit - in dynamic relations to its environment. On the other hand, it is clear that the outer workings of an organisation pertain to its purpose and manner of function within its environment, while its inner working pertains to the manner of implementation of these externally defined purposes and functions. This view suggests a technical possibility that I believe is in the midst of becoming a reality: a method of system description and analysis which applies equally to system purposes and system implementation. Such a method would do much towards establishing a firm link between the two - a link to be devoutly desired, for reasons that hardly need belabouring, in this company.

Another way to characterise the applied field we are talking about is by its practical problems. "A system is" so one might say "a context in which the following sorts of problems arise". To this end, we now provide a sample list, in no special order:

System problems: Outputs or inputs of wrong content, wrong timing, wrong location; bad side effects - such as security leaks; inconsistencies resulting in deadlocks, endless looping, critical races, unbounded delays, unbounded storage requirements, unresolvable conflicts; inability to back up in case a part fails; unanticipated and unacceptable resource demands at some times and places; one harmless - seeming change in time, place, procedure, format, etc. resulting in many harmful consequences; having to stop everything in order to repair one part; having to change everything because of a seemingly small change in functional requirements.

Problems of these types may arise in the context of any organisation, whether or not computers are employed, but they will tend to arise more frequently, insistently, and ferociously in contexts involving computers. Therefore, it is these contexts most especially which make urgent the demand for effective theories about systems -- theories about systems, rather than theories about computers. From this point of view, computers should be thought of as physical structures that are a part of particularly sophisticated and elaborate organisational mechanisms, just as manual files, ledger books, mailing machinery, cash register, calenders and clocks are physical structures so employed.

All such physical structures require a context of rules-of-use (who, when, where, and how) which make them a part of organisational mechanisms, and this is true of computers just as it is true of these simple devices. Theories which focus our attention on the operation of machines independent of the purposive organisational context of their employment do us a disservice in several nameable respects.

Such theories create a conceptual wall between the description and analysis of these contexts and the description and analysis of the machines. There are no walls as impregnable as conceptual walls.

In organisational contexts, the effects of decisions that result in action propagate from people to people, from people to machines, from machines to people. Such theories extinguishes the hope for a theoretcial model which permits us to trace the flow of these effects -- one of the principle powers which an adequate theory of systems must deliver.

Such theories make it difficult to understand in what technically adequate sense organisational functions carried out by people can instead be delegated to machines.

Shifting from negative to positive accents, let me talk about necessary and desirable properties of a theory - or theories - of systems in the spirit of the above.

To begin with, such a theory should provide axioms that relate a set of widely applicable system primitives to one another - primitives applicable to the description of people, objects, and machines, fulfilling organisational functions.

We need to develop such primitives for many reasons - for instance, to promote the transferability of intellectual products within our field. A simple organisational invention (such as semaphores) may find its concrete application in programming, in computer hardware, in railroad traffic management, or in office procedures. Similarly, a new mathematical technique for the analysis of deadlocks should become applicable in all of these various settings. We also need such primitives to develop a basis for common understanding among the many different classes of specialists - designers, operators, managers, users, implementers, lawyers, accountants, etc. - all of whom must deal with systems, each from his own point of view.

Secondly, an adequate theory in this field must have a substantial mathematical arm. That is because one of the main contributions that theory can make to systems practice is to offer calculation aids to the solution of problems, such as those listed above.

Our theory must permit the description of systems at many different levels of detail in a uniform manner, with formally defined connections between different levels of description of the same entity. That is, I believe, a necessary part of establishing the relation between system, implementation and purpose - as already discussed above - and it is also necessary for the capacity to describe complicated systems clearly and concisely.

Our theory ought not to rely on the fiction of instantaneous state change. The coming about of an objectively verifiable change in anything always takes time, and necessarily entails passage through a region of uncertainty. Our theory should also not pre-suppose the fiction of perfect classification schemes - classification schemes which eliminate the <u>possibility</u> of in-between cases in reality. Theories based on these fictions cannot account for the effort that must go into achieving reliability in systems, nor can they contribute to an understanding of the limits on agreement as to the time of a change or the value of a variable at two distinct places.

Our theory must provide a technically adequate explanation of what it means for one decision to influence another. Technical adequacy is to be demonstrated by providing calculation tools that enable one to trace the effects of decisions with the help on one's system model. This ability is critical to many practical systems tasks, such as debugging, designing back-up capabilities or security measures, specifying repair procedures, and many others.

Our theory must provide the conceptual machinery for the translation back and forth between requirements on system operation expressed in relational terms - for example, the amount of money withdrawn from an account must be less than the current balance, the operator must receive permission to turn power on before he does - and requirements expressed in terms of parameter values, formats, location designations, etc. - for example, the second field of the record representing an account must contain a five digit decimal number representing the current balance, permission to turn power on must reach the operator by 8.00 a.m. every morning, etc.

Why is the development of adequate theory about systems important?

Perhaps the most pressing reason in our society is the advent - not to say onslaught - of computer and communication technology. These technologies produce enormous organisational change in our society in very short spaces of historical time. Since, in the past, major organisational mechanisms - such as transportation systems, fiscal systems, information storage and transmission systems - could only change a little bit at a time, the processes of gradual evolution and natural selection could help safeguard against disaster. These safeguards are no longer with us. The ability to understand our organisational mechanisms and their effects has become an urgent social need.

It is also true that in the presence of these new technologies, there are long and complex chains of organisational influence which are mediated entirely within electronic communication and computing links - and over these pathways, the review of human common sense cannot be exercised. Common sense cannot be automated, no matter how much chess playing can.

An adequate theoretical and practical understanding of information and its flow is desperately needed in the formulation of intelligent public policies about information - its ownership, generation, transfer, validation, destruction, etc. Public policies, as expressed in legislation may do little good if the actual machinery with which society carries on its business makes the intent of that legislation unenforceable. How to build information handling machines consistent with the intention of law, and how to design laws consistent with existent machinery are non-trivial problems demanding, I believe, technical understandings of the sort I have been describing.

It is my belief, that our field has suffered less from a shortage of technical ideas that should be part of a graduate training in computer science than from a shortage of technical ideas that should be part of every high school student's education. The formal tools he now acquires in his courses in mathematics, physical science, and computing are, so I believe, fundamentally inadequate to giving him the means to represent the realities of organisational mechanisms and to manipulate these representations in trying to understand the consequences of these mechanisms. Yet these mechanisms as deeply affect the life of everyone as any other aspect of reality. It is my earnest hope that Petri-nets will help to fill this gap.

Remarks on the State, Direction, and Correction of Computer Science

The following is a set of remarks I made to the Principal Investigators Conference in Los Angeles, February 6 - 8, 1974, after hearing Steve Lukasik on his dissatisfaction with the program of the Computer Science community towards helping the Department of Defence with its basic objectives.

I, as a member of this community, have also been dissatisfied with the general progress of Computer Science during the last decade and, as I suppose is natural, tend to see this dissatisfaction as related to my own. I feel that the issues are of major importance to our field, and I make my remarks, not as an outsider throwing rocks, but as a dissident member of a group hoping to exert some small influence upon his fellows for the sake of a common good.

A brilliant chess move as such, is not, in my view, an exhibit of intelligence. For instance, a brilliant chess move while the room is filling with smoke because the house is burning down does not show intelligence. If the capacity for brilliant chess moves without regard to life circumstances deserves a name, I would naturally call it 'artificial' intelligence.

I believe that we, as a community, have been in the grips of artificial intelligence. The house has stood in flames for quite a few years while we have been busy playing chess. Furthermore, while the Lukasiks of this world were pouring money on us because of their expectations of good things to come, many of us have become accustomed to the life style and mental habits of an aristocracy. Nothing is too expensive if it will serve us: another 10^5 words of high speed memory, another few CPU's etc. 'It is people's time which is expensive' one hears in connection with such proposals for acquisition, so the cost of the equipment is small by comparison. The 'people' to whom one is referring in that sentence is 'me and my kind', and the concern one is showing has little to do with saving somebody's money. All that is meant is: there is an absolute scarcity of the good results which I, the computer scientist, can produce; and if larger memories and more CPU's can increase my productivity, it is worth it to society. I proceed from the assumption that, apart from food and shelter, there can be no absolute scarcities. Insofar as the production and maintenance of large scale computing capacity involves large scale social effort, the plea for more of it is a plea for something in the face of competing social claims. To pick an example near to home: the Department of Defence has to buy not only computers, but also ammunition. In this context, the obvious reasonableness of the claim to more memory and more CPU's evaporates. What I'm getting at is this: although the sentence 'It is people's time that matters' sounds humanitarian and service-minded, I think it is an expression of strictly self-serving attitudes.

Lukasik said yesterday: you guys have not been of service to your paymasters. I think there are many others to whom we have also not been of service. To pick a random example: we have not been of service to me. In what I do, computers do not help me. That is not because they couldn't; they don't. The reason they don't is because computer builders have not found out what I can use that is mechanically implementable.

Let us consider some of the aspects that are critical in being of service to someone with machines of the type that we build. The machines have buttons (or keys) which can be pushed by someone who wants something to happen (or not to happen). I would take this to be true even if the button were "pushed" by voice. Now the following is critical, if the possibility of pushing a button is to be valuable to a user - for example, a general:

A. The general, when he pushes the button, must know how his situation has changed as a result of that action. In other words: he must know what he can safely do (or not do) before he pushed it.

For condition A to be satisfied, the button-push must "factor in" to the structure of his organised activity. That is one of the important keys to the success of calculators. Important segments of organised activity are expressed, or can be expressed in terms of operations on numbers. That was true before electronic calculators appeared on the scene. Thus calculators fit naturally into pre-established positions in the flow or organisational functioning.

Another important aspect of the success of calculators has to do with the issue of safety, referred to in condition A. No positive action is ever absolutely guaranteed to have its intended effect. But one always acts in the light of specific expectations of failure. For instance, if I ask someone to pick up a package at noon tomorrow, I realise that he might forget. If it is important enough, I might take the additional precaution of arranging for a timely reminder. I do not, however, as a precaution, lock the door to my office for fear that, as a result of a failure in the carrying out of my instruction, the messenger will attempt to paint my office walls. Here again, calculators are successful because already established modes of protection against potential failures in the absence of calculators continue, more-or-less, to be effective in their presence - for example, doing the same calculation twice before taking the next step which makes use of the result.

Now, in order to help someone (or a class of someones) by the introduction of sophisticated machines, it is absolutely critical to understand the structure of the organised activities that are to be improved. This understanding is not easy to come by because typically the man who plays the role of carrying on these activities does not understand them himself from this point of view. That is natural because his goal is not the fitting of machines into his patterns of action but rather, his goals are those which are intrinsic to those actions. Thus, he cannot himself be directly helpful in the acquisition of the understanding to which I have just referred. This, in my opinion, is the important content in the common remark that the general cannot say exactly what he needs or expects from the systems that might be built for him. That does not mean that no one can say. In particular, we must be able to say, if we are to succeed with helping him - and we must be able to say it in such a way that the content of that saying can be converted into a set of design specifications. Only insofar as we are successful with this analysis of human activity in an organisational context will we ever succeed with the introduction of complex computer-based systems which, for its users, satisfy condition A.

The main trend in attempts to build usable computer-based systems has been based on a different conclusion from the basic fact that the general cannot say exactly what he demands from a system. We have tended to conclude that there simply isn't anything exact to discover. To me, this is analogous to the following: upon asking a speaker of English about the grammar of his language, and discovering that he knows nothing about it, one concludes that English has no grammar. Once one accepts that the general cannot be expected to mean something exact when he pushes a button on a console, one concludes that, to help him, one must build machines capable of "understanding" vague instructions. I believe this to be a bad conclusion (ultimately incompatible with the satisfaction of condition A). Whether one agrees with me or not, it is clear that in the computer science field little attention has been paid to the problem of relating the structure and function of human organisations to the structure and function of the machines that are supposed to fit into them. I am not talking about human engineering which concerns itself with the contact surface between the man and the machine (and not with the structure and function of the machine) and I am not talking about management science which accepts the machines that exist as tools to be used as best as possible. I am talking about <u>the endeavour to understand our information machines as expressions of organisational intentions</u>. To make progress on this front requires that computer science recognize the relevance of the study of human organisation to the understanding of information processing in its by-now traditional computer-science sense.

It is not my purpose in these talks to concentrate on the theoretical work which my colleagues and I have done in the last ten years. Nevertheless, in order to develop intelligently and intellibibly the rest of what I wish to communicate to you, I must give some introduction to the basic concepts in terms of which I have learned to think. These concepts are intimately connected with the study of Petri-nets.

Petri-nets may be viewed as a family of formalisms for systems analysis. The work on Petri-nets may be broadly divided into two types: (1) mathematical and (2) semantic.

Mathematical work includes: specification of axioms, proof of theorems and finding of algorithms. Semantic work involves the development of applied concepts which, once learned, can lead a system practitioner to formulate his system understandings in structural forms which satisfy axiomatic requirements. Successful semantic work leads to the analogue of the ability to translate work problems into systems of algebraic or differential equations to which the formal methods of these branches of mathematics are applicable. Let me give you an example of a simple prototypical work problem in systems analysis, relative to which we already had five years ago, the complete machinery - the semantics for translation into formalism and the mathematical machinery for algorithmic solution.

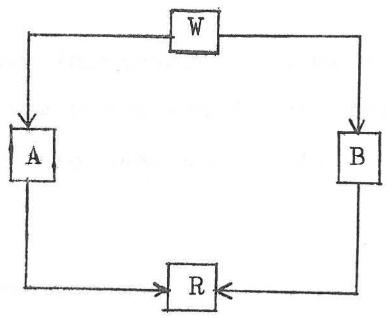
A reader R and a writer W are connected to one another by two buffers, A and B as shown in Figure 1. As long as R's activity continues, he strictly alternates between reading buffer A and reading buffer B; as long as W's activity continues, it strictly alternates between writing to buffer A and writing to buffer B. Each of the buffers alternates between being full and being empty. If R becomes ready to read a buffer when the buffer is empty, R waits until it becomes full; as a result of reading the buffer it becomes empty; if W becomes ready to write to a buffer when it is full, W waits until it becomes empty; as a result of writing to it the buffer becomes full.

Questions:

1. In how many ways - 0, 1, 2, ... etc. - can deadlocks arise?
2. If the system is set to an initial condition in which a read, or a write (or both) is possible, can a deadlock arise after some number of reads and writes?
3. The purpose of having two buffers rather than one is to permit concurrent reading and writing. Do initial conditions exist which defeat this intention - that is, guarantee that on and

Problem Conditions

1.



Writer W and reader R connected to each other by two buffers, A and B.

2. R and W both alternate between buffers A and B.

3. The buffers are either full or empty.

4. R can only read a full buffer; W can only write to an empty buffer. Reading to a full buffer empties it; writing to an empty buffer, fills it.

Questions

1. How many distinct deadlock configurations (if any)?

2. Starting with a configuration which enables at least one action (read or write) can a subsequent pattern of actions lead to a deadlock?

3. How many initial configurations (if any) which guarantee no concurrent reads and writes?

4. Starting with a configuration which enables a concurrent read and write, can some subsequent pattern of actions lead to a configuration which guarantees no further concurrent reads and writes?

Figure 1

only one action - read or write - will ever take place at a time? If such initial conditions exist, then can every such condition be reached from every other such condition by some pattern of reads and write?

4. Starting with an initial condition in which a concurrent read and write can take place, can some pattern of reads and writes lead to a condition in which only one action can take place at a time?

The above qualifies as a meaningful exercise for someone who is learning to analyse systems.

I will now give a list of fundamental concepts in relating systems (like the above) to Petri-nets. We have:

1. Role, activity.
2. Actor.
3. Condition (state, transition).
4. Behaviour (or an actor, of the system).
5. Status image.

This is in no sense a complete list; it is all that we have time for. What I hope you will gather from the following discussion is: (1) that these concepts, though natural once seen are not abvious; (2) that they capture basic aspects of systems - aspects which any theory with the objectives sketched above must somehow deal with, and (3) that they fit together elegantly in the Petri-net context - elegantly enough for mathematics. We must however leave the discussion of the list until the following session.

Discussion

Professor Randell opened the discussion period by asking whether there existed other methods for representing systems, which might be rivals to the Petri-net theory. Dr. Holt replied as follows: 'There are many methods which, in some sense, deal with aspects of systems, as here understood - graphic methods like PERT-CPM, net-work flow analysis (Ford/Falkerson), logical calculi (Henrik von Wright) etc. At this stage of development, Petri-nets have seemed to me like a particularly useful vehicle for bringing systems as a proper object of scientific study into view - but the important thing is the subject and not Petri-nets. In the present embryonic state of development of the subject itself, it is too early to talk about significant rivalries among methods, since no method, as far as I know has reached any great stage of advancement'.

The only other question came from Professor McConalogue, who asked if instantaneous clearing of the buffer in the reader-writer problem was important. Dr. Holt felt that the speed of emptying the buffer was not crucial to the problem, as he had posed it.

Lecture 2

I would like to explain what I mean by role in an organisational sense. People and objects may be actors who (which) play in an organisation (see Figure 2).

The ideas of role and actor are fundamental to the understanding of this subject; also the student must gain expertise in use of these terms in applications of this method.

The effect of participation in an activity on a role player is a change in his state. (This is another formal word.)

Figure 3 shows the application of the concepts role, activity, state to the description of the Reader-Writer-Buffer problem (see the informal description of the problem in the previous lecture).

Note that the effect on W of writing to buffer-A is a change-of-state from having a record for buffer-A to being ready to generate a record for buffer-B.

The concepts developed so far (actors, roles, states of actors and activities) can now be applied to Petri-nets.

A first approximation to the formal representation of a role is a state transition graph with labelled arcs and vertices. The labels on states indicate their meaning and the labels on arcs indicate which activity results in the state transition represented by the arcs. We wish to recognise the activities as primitive entities like the states and therefore represent them as new vertex types.

Figure 4 shows this representation. A state transition graph is the possible behaviour of an actor playing a given role and are represented as the state transition sequences which are generated by the role description.

Figure 5 shows how to view the sequential behaviour of an actor who is playing a role. The figure shows two interval-types available and engaged, in alternation. The two step-types which result from the sequence of two interval types are called commitment and completion.

(Organisational) Roles and Actors	Examples
1. People and objects, separately or in combination, may be actors that play roles.	1. Role examples: reader, writer, buffer (recall the "word problem") record, file, buyer, seller, chairman, president, ledger, etc.
2. One actor may play several roles at one time, and/or in succession.	2. One actor, several roles example: It is part of the duty of the president to chair the stockholders meeting. At the time of this meeting, there is one actor playing the role of president and the role of chairman.
3. One role may be played by several actors - at one time, and/or in succession.	3. One role, several actors example: A stream of records pass over a sequence of processing stations. All the concurrently present instances of records in the stream are actors playing the role of record.

Figure 2

<u>Activities of the Writer W</u>:	1	Writing to buffer A
	2	Writing to buffer B
	3	Generating next record for buffer A
	4	Generating next record for buffer B
<u>Activities of the Reader R</u>:	5	Reading buffer A
	6	Reading buffer B
	7	Using the record from buffer A
	8	Using the record from buffer B
<u>Activities of Buffer A</u>:	9	Being written to by W (same as 2)
	10	Being read by R (same as 5)
<u>Activities of Buffer B</u>:		Analogous to buffer A

<u>The effect of writing to buffer A on W</u>:

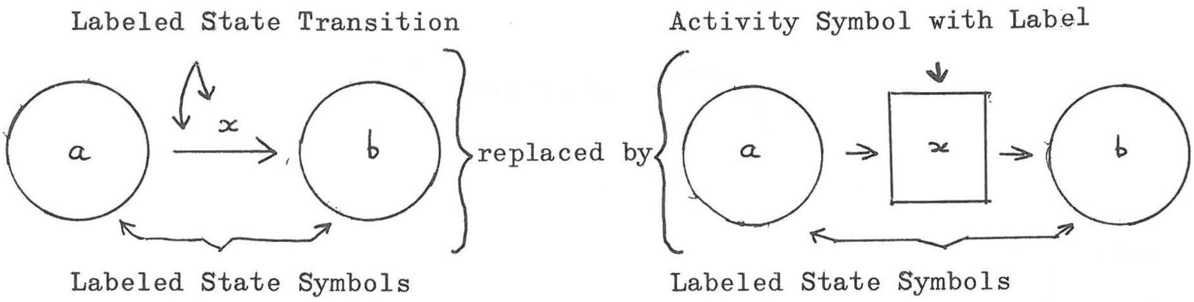
A <u>change-of-state</u>: from having a record for buffer A to being ready-to-generate a record for buffer B.

<u>The effect of the same activity on buffer A</u>:

A change-of-state from being empty to being full.

Analogous descriptions apply to the other activities in this example.

Figure 3 Example of activities, states and state changes based on the Reader-Writer-Buffer problem.



Now we can represent:

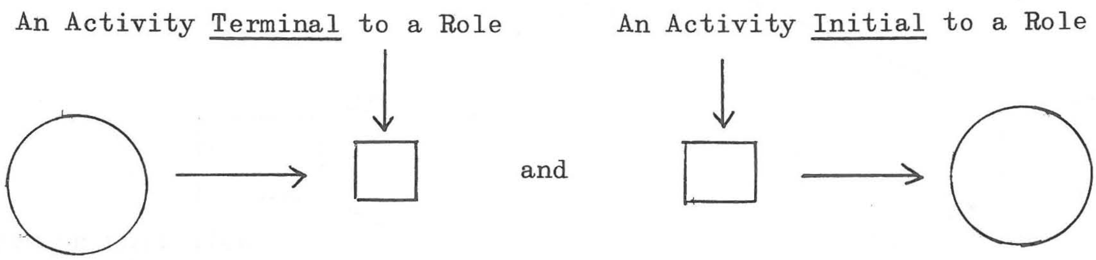


Figure 4 Representation of transitions

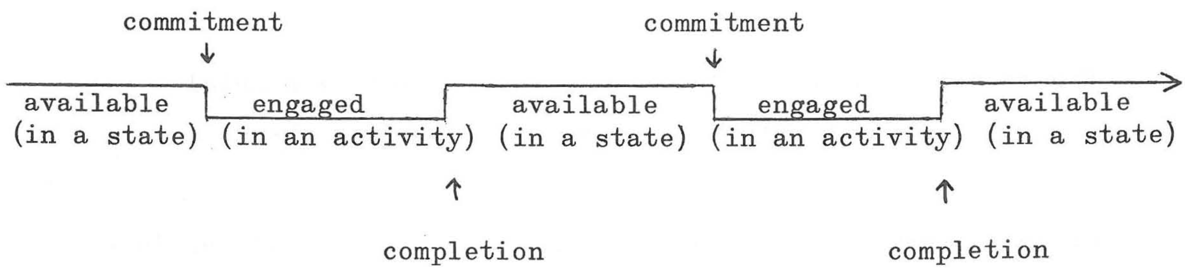


Figure 5 Life History of an actor playing a role

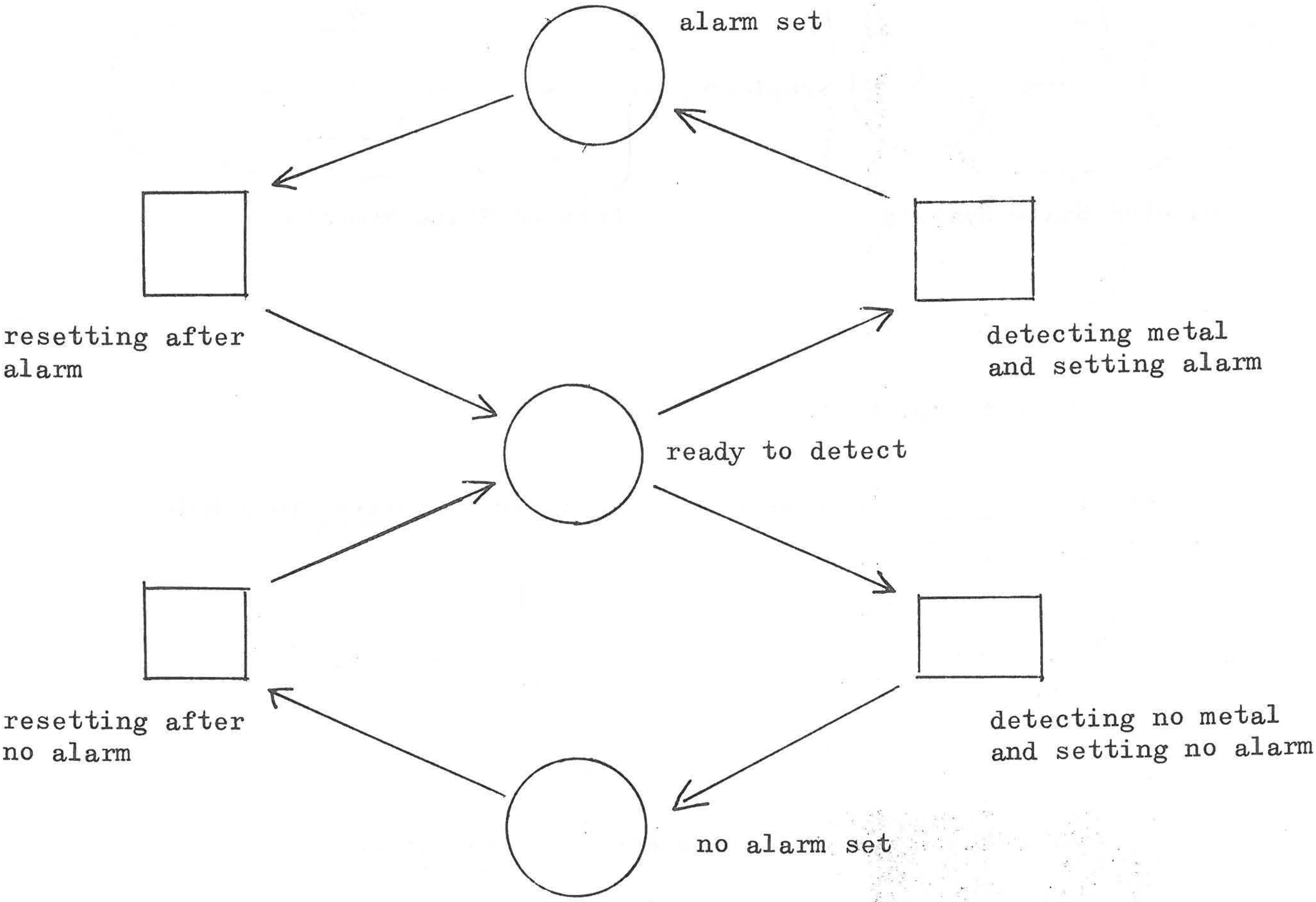


Figure 6 A gross-level role description of a metal detector for an airline security check

Figure 6 shows an example of a simple role-a gross-level description as a metal detector for an Airline Security check. There are three states in this model. The meanings of the four activities in the figure are as shown by their labels. These activities are necessary for fulfillment of the function. To understand these meanings one must consider the typical situation in detail and examine the real life implementation of the activities shown. On being questioned by Professor Cole about this figure "How does the detector device know it has to be re-set after there is no alarm?"

Dr. Holt answered as follows: Assume the detector operates by buzzing when near metal and not buzzing otherwise. Now suppose the detector has been brought into the proximity of a suitcase and then into the 'ready' state for detection; it now either buzzes or not. Its failure to buzz must be noticed, just as its buzzing must. The implementation of the speed with which the detector responds to the presence of metal, and on other matters which we pass over here. After the detector has buzzed or failed to buzz, it must be removed from the vicinity of the suitcase before it can be used for another detection. Professor Dijkstra asked if one might distinguish, in principle, between negative and positive activities ('no alarm' as example of the negative). Dr. Holt replied that an adequate treatment of the topic of the negative both with respect to states as well as events is beyond his present capabilities.

Activities usually involve the interaction and co-operation of a set of roles. As an example of this, the model of the four roles, reader/writer/buffer-A/buffer-B, will be examined again. Figure 7 shows the Petri-net which connects these four roles to one another. There are four states in the top of the diagram; these represent the role of the writer. There is a write to Buffer-A and to Buffer-B. Both of these steps change the status of the writer and require co-operation from the appropriate buffer. (The change of states are shown by arrows. In general, there may be other activities which are connected to the model, but these are ignored here for simplicity.) The status of the buffer changes by going from empty to full and the writer, as a result of the same activity changes from having a record available for A to being in a state prepared to generate a record for B. In the generation of that record for buffer B there will be other resources required, but these are ignored. We will only consider the boundary between the writer and the rest of the system. The coordination of the activities between buffers A and B, and the writer, is now clear. The reader activities follow a similar pattern where the reader co-operateswith buffer-A and buffer-B activities.

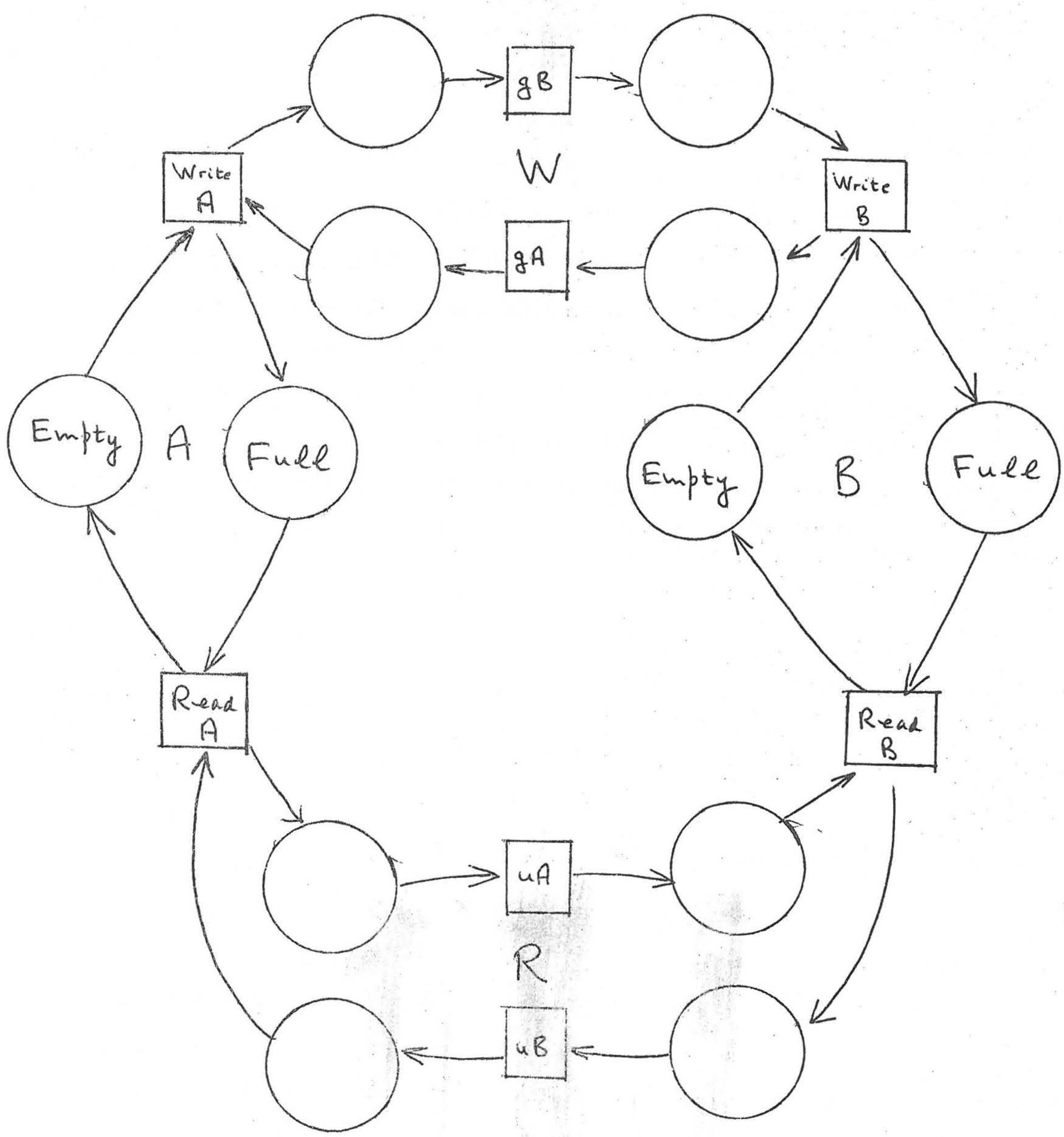
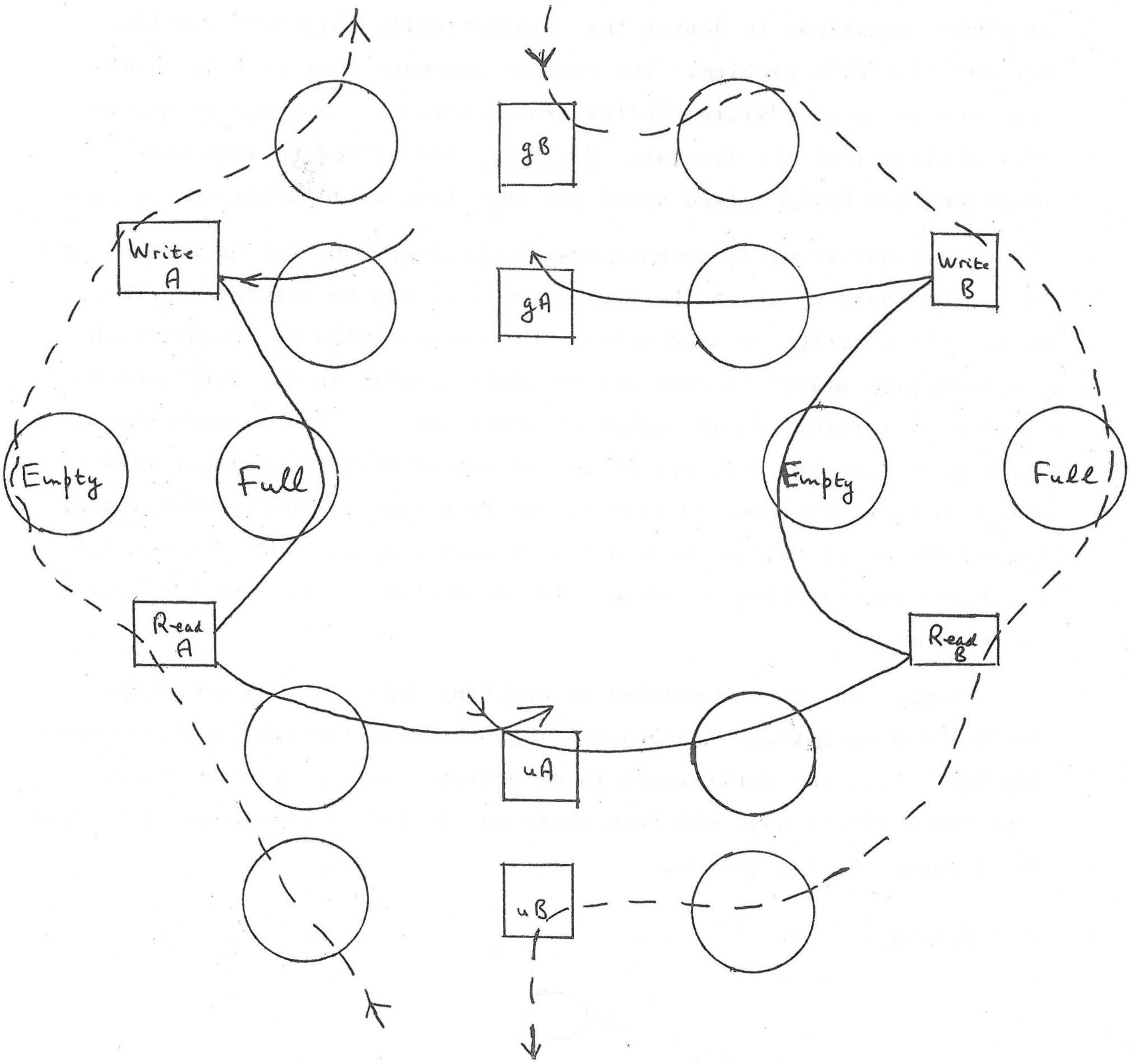


Figure 7 Reader-writer-buffer problem conditions translated into Petri-net form.



—— indicates primary flow

- - - indicates secondary flow

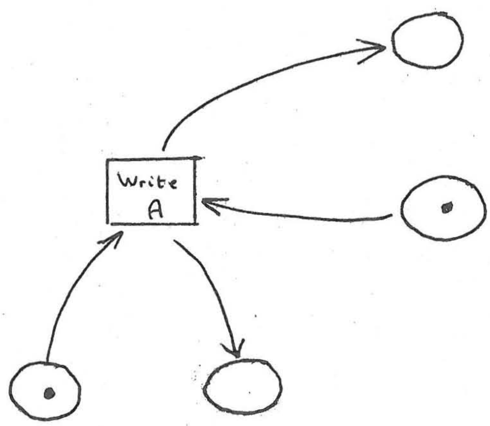
Figure 8

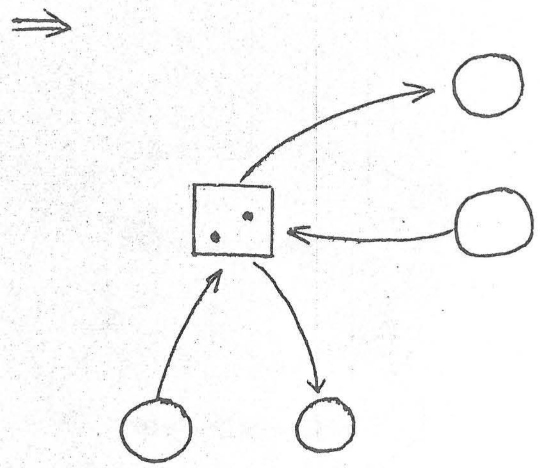
The figure is the equivalent to the setting up of a system of algebraic equations to depict the mathematically relevant semantic content of a work problem. The comment was made that writing equations for the reader/writer/buffer system would be an order of magnitude smaller that the diagram. Dr. Holt said he had no idea what equations are being talked about and why they would allow one to see.

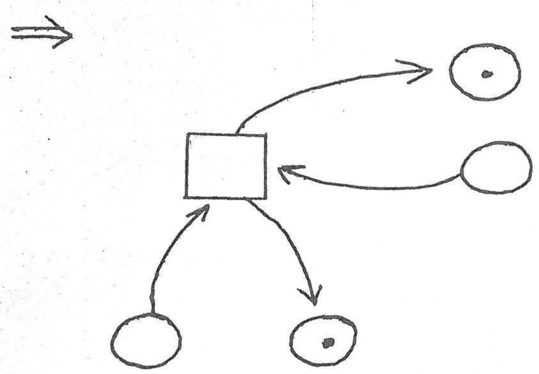
Petri-net figures, such as the one in Figure 6, may be thought of as a game board on which the system activity can be simulated. The actors who play the version roles can be represented by tokens which are initially placed on some set of state symbols in the Petri-net to suggest an initial configuration of actor states. In Figure 6 the state partition by role, and it is understood that one-and-only-one actor is to be assigned to each of the four roles. Thus, only initial distributions of tokens which place one-and-only-one token on the set of states representing a single role can conform to the intended class of meanings.

Simulation can now proceed by enacting some activity which the initial configuration of state-holdings enables: the tokens representing the actors who participate in an activity can be jointly pushed into the activity box, and from there out to the circles which represent their post-activity states.

Example:

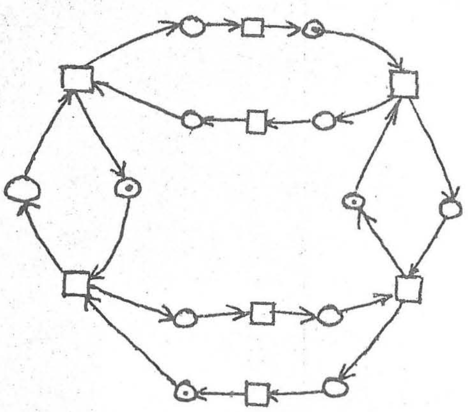




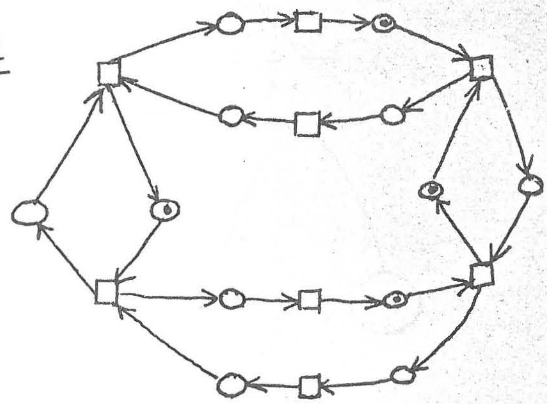


In this manner the simulator can progress from state-configuration to state-configuration on the Petri-net. These configurations are traditionally called *state markings* - or just *markings*, when the configurations in which tokens rest in activity boxes are not counted as of special interest. The collection of markings connected to one another by simulation steps are called marking classes. The questions posed in the reader-writer-buffer problem are questions about the properties of marking classes of Figure 7. The following four markings of Figure 7 belong to four disjoint marking classes.

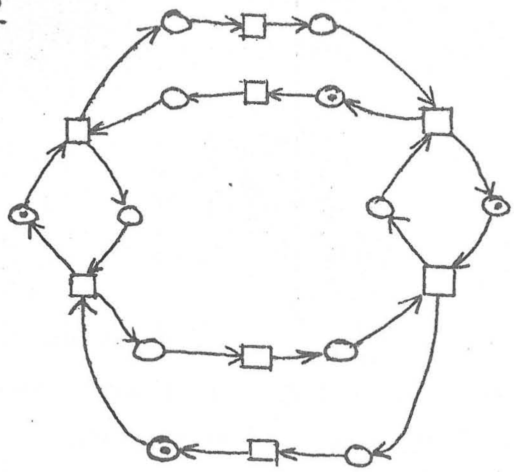
1



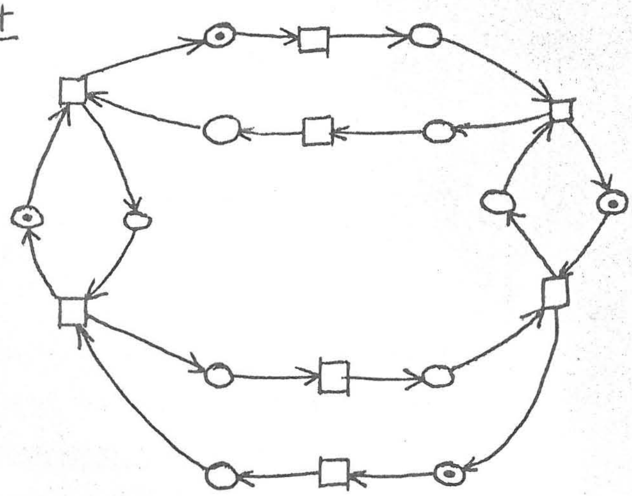
2



3



4



The marking class to which 1 belongs permits concurrent reading and writing; the marking classes to which 2 and 3 belong represent patterns of activity in which all activities are strictly sequential; the marking class to which 4 belongs contains deadlocks. Properties of marking classes such as those above can be determined to hold or not hold by algorithms which examine only a single specimen marking from a marking class. Such algorithms are the basis for answering questions, as in our sample problem. (Given two specimen markings from a marking class, one can also determine whether they do or do not belong to the same marking class.)

While Figure 7 and the above discussion illustrates the kind of analysis which our representations of systems should enable, Figure 8 illustrates the enhancements in semantic understanding to which our representations should lead. It shows that the definition of the activity relations between the four roles - reader, writer, buffer-A, buffer-B - naturally leads to the recognition of four new entities in terms of which to understand the interactions - items which flow through buffer A, items which flow through buffer B, and items representing permissions for the forward flow of the just named item types - any implementation of the forward flow items without collusion will always involve the implementation of a "back-flow" of permission, as illustrated in this case. Further discussion of this topic is not possible on this occasion.

The purpose of the foregoing example and its somewhat fragmentary discussion is to convey the flavour of those concerns which in Dr. Holt's opinion, should play an important role at the beginnings of system science and at the beginnings of an education about the function of computers.

Lecture 3

It is Dr. Holt's belief that the system subject has special relevance to communication technologies. These he understands to include technologies for the transmission, storage and transformation of messages. It also has relevance to technologies for the transmission, storage and transformation of tangible goods, and thus could be constructed as relevant to all technology. In neither case - messages or tangible goods - does it address all technical problems with which these technologies must deal: it is more relevant to technology concerned with messages because it addresses a larger fraction of the significant problems in this case. Instantiated messages may be thought of as physical objects, whose systemic properties are the largest part of what matters about them (this is not true of all messages, for example poetry, but is true of messages in a formal context).

What of computers and computing, asks Dr. Holt? The systems point of view would he says, principally see computers as communication machines and principally see the material that flows through a computer as documentsbearing messages. This means, among other things, that the items flowing into and out of computers must be understood, so Dr. Holt says, as having the attributes of messages - such as sender - or a group of senders - a receiver - or group of receivers - an expiration, authorisation, validation, copy ID, etc. These various attributes must be understood to be present whether or not they are represented explicitly by bits. They may, for example, pertain to an item by virtue of the spatial domain to which it is confined and the time or times when it is to be found at one location or another. In any case - so this point of view would suggest - the significant aspects of computing functions - as abstractly defined or concretely implemented - are not grasped without accounting for the message attributes of items. Dr. Holt here displayed Figure 9, which is a table of several concepts of a computer in use, including the one he just mentioned, and said that thinking of the entries in this table as interpretations of computer activity one might ask: is there a principal interpretation in terms of which the essential features of all applications may be comprehended? He thinks the answer is 'yes' and that most people also think the answer is 'yes', but that most people and he disagree about what that principal interpretation might be. The widely accepted answer is 2 in Figure 9, while Dr. Holt's answer is 1 and he went on to give arguments to show that these two interpretations are not simply the same thing described in different words.

The identity of a function depends upon the identity of the elements in its domain and range but not on how they happened to be named. In performing an evaluation however, the argument value treated as input, and the function values, treated as output, must be named somehow since the evaluator cannot deal with disembodied elements. How the elements are named is, in general, not a matter of indifference. The user (or users) will have definite requirements as to the form of the output names, while the evaluator - that is computer and program - will have such requirements on the form of the input names.

Thus, in addition to the evaluation proper, there may arise the necessity for the recoding of element names on the occassion of inputting and/or outputting. These recoding operations would ordinarily be seen as auxiliary to the principal step - the evaluation proper - and of lesser importance from the computer theoretician's point of view.

But now the question arises, what is the point of evaluating a function in the first place? If the value of f(x) for the argument value c is to be found, why is the expression f(c) not normally regarded as a fully adequate answer? After all, it is a designation of the desired value.

The only reason that Dr. Holt can think of is that, from the end-user's point-of-view, this answer is in an <u>undesirable form</u>, and what this means in turn is the following: that, with the resources available to the user, the answer in the wrong form does not permit him to take the next step in the pursuit of his purpose, the next step which depends on the receipt of the answer.

The Role of the Computer	Example of its Input	Example of its Output
1. Message processor	Messages	Messages
2. Function evaluator (e.g. numeric calculator)	Argument values	Function values
3. Process controller	Data	Control actions
4. Problem solver	Data and problem	Answer

Figure 9

But that is the very same reason why steps of recoding might have to be undertaken upon the occasion of inputting argument values or outputting answers. (Upon input the issue is to satisfy the requirements of the evaluation machines - computers and program - in the light of the organisation of its resources.)

There are various aspects which arise to notice in the definition of an event of input or of output, such as:

1. The actual coded format of the information.
2. "When" it takes place.
3. "Where" it takes place.
4. "From whom" (or what) or to whom (or what) it is directed.

These same aspects remain relevant in the performance of functions internal to the machine, but relevant in a different sense. At the boundary (input or output) they are part of the <u>specification</u> of what pattern of use is to be implemented. Internally they are relevant because of the manner in which the computing resources have been packaged, their relative speeds of performance and their relations of interconnection.

If one's theory does not take into account these various aspects at the theoretical level it has two effects:

1. It cuts you off from being able to express in theoretical terms what in fact is the function that the piece of machinery is supposed to perform.
2. Internally tends towards a separation of issues of "where" and "when" and how fast etc., from the issues of what the actual content of transformation is.

These are both, in Dr. Holt's opinion, bad effects.

Notice in passing that one cannot think of these aspects of context to be theoretically replaceable by expanded character content of the input or output - characters which designate the properties of input/output named above. That is because the expanded item would nevertheless have to be coded according to some convention, appear at some place and time, etc. There is an exact analogue to this in ordinary communication, for example it is clear that when somebody communications there is an aspect which is relevant, which one might call tone-of-voice, and of course, the tone of voice communicates. Could that tone of voice be replaced by words? Dr. Holt says that in principle it could not, because after it was replaced by words it would still have to be said in some tone-of-voice. Professor Hamming here asked about the case of the written card, and Dr. Holt replied that it is a mistake to think that written words do not have a tone of voice, for every piece of writing has an associated tone of voice by which Dr. Holt meant the publisher under whose name it is published, the kind of title it has, the shape of the pages on which it is printed, the style in which the illustrations are drawn etc., which all make a huge difference to the way in which those words are interpreted. Dr. Holt was then asked whether the concept of number can exist without a representation and he replied that a mathematician would say 'yes', but that he thought this was a bad point of view in the field of computing.

Dr. Holt continued by saying that the various aspects of the definition of an input/output event mentioned above naturally obtrude themselves on one's attention if one is thinking in terms of documents bearing messages as distinct from thinking of values in the sense of mathematics. For example, no one is tempted to think of a typed presentation of a paragraph in French and a typed presentation of a paragraph in English as instances of the same document, even if the French paragraph is a translation of the English one. The aspects do not, however, obtrude themselves on one's attention if one thinks in terms of values in the matematical sense. While conceptually separating the function that is to be evaluated from the communicational context in which the evaluation takes place has its use in certain instances of computer application - for example those instances where mathematical knowledge of the relation between functions can be used to affect implementation - it is, Dr. Holt believes, an inadequate intellectural method for dealing with computers and computing in general.

Conventional	Systemic
Information is an imperishable good.	Information has validity only within a given context.
Information content and form are in principle, factorable from one another.	Form and content are, in principle, inseparable.
Information can, in principle, flow from sender S to receiver R without information flowing from R to S.	Information flow requires, in principle, circuits over which to flow (like electric current).
The information which R receives can, in principle, be the same as the information which S sends.	The opposite i.e. it cannot be.
In communications and computer engineering we need not concern ourselves with the meaning of the information that is processed.	The opposite.

Figure 10

Dr. Holt now went on to discuss concepts of information, and at this point displayed Figure 10 to show the relationship between two types of thinking about information. He then went on to say that, related to the concepts of what a computer does - or for that matter what a telephone does - are, of course, concepts of information. The predominant view of information is as an imperishable good - in analogy to the predominant view of gold. This fits well with the idea of computing as function evaluation. The domain and ranges of mathematical functions are not prone to decay. Conversely, when new domains and ranges are defined one does not think of these domains and ranges as having freshly come into existence, but rather of their being freshly noticed. The real numbers or the integers are prototypical of imperishable goods.

This concept of information is applied in practice to the organization of libraries and data banks. That information, in practice, does not conform well to the model leads to the awkward problems of obsolescence for libraries and data banks - i.e. coping with information obsolescence (related to the policies of acquisition in the face of storage space limitations) and the development and updating of effective information classification schemes.

An effective information store cannot be designed without regard to the <u>transitory</u> character of all <u>useful</u> information, and hence in the light of specific expectation as to rates and kinds of change in information content and accessing paths.

Dr. Holt then went on to mention some of the social implications of computer function concepts, and gave a list of these.

1. How people see themselves and their own function in relation to machines and their function.
2. What aspects of computers and other communication machines will be seen as making important differences - for example not only, how easy is it to make information items interact with one another, but also how easy is it to

insulate information items from one another, so that they will be guaranteed not to interact; how computers interrelate with telephones and other information transmission systems.

3. In what terms to formulate public policy, partly to be expressed in legislation, governing the construction and use of computers.
4. How to approach the definition of a computer application. What are the critical parameters of an application?

At this point Dr. Holt talked about items not to be omitted in learning about computers, and he gave the following list of points. This list, he pointed out, makes short shrift of those matters which are already at most computer science educator's focus of attention - such as how to teach programming. Dr. Holt agrees to the importance of these matters, but wishes, with the help of the list, to draw attention to other matters which he believes also need attention.

1. System concepts: to enable one to understand computers and programming from the systems point of view.
2. Experience with programming and computer building. Dr. Holt said here that there had hardly been a mention of building during the conference.
3. The structure of the computer and computing business, and how it got to be that way. (By the structure of the computing business, Dr. Holt meant the way in which it is organised - software houses, consultants, companies of various sizes etc. In answer to questions from Professor Page, about whether this view would have a quantitative aspect, Dr. Holt said that his intention would be to be true to facts, but that facts are given bias by the way in which they are presented.)
4. What is the state-of-the-art in the various sections of the field.
5. Concepts of computer use: common and technical.

(By "common" Dr. Holt implies the way in which people talk about computers, e.g. "the giant brain", "the super inteligence"; these matters do not have technical content, but are a reflection of the way in which people regard computers.)

The reasons for the development of these usages, and their practical effects.

6. Concepts of Information: common and technical.
(This would include, on the technical side, such things as semantic theory, the theory of coding and representation of information and the ideas that have come from Dr. Holt's work.)

7. Social Issues:

Concepts	Practice	Regulation
Person/computer, communication and what it depends on.	security privacy Obsolescence of information and systems reliability responsibility	legislation professional ethics

(Professor Page here pointed out that legislation, and professional ethics would vary greatly from country to country, which could lead to some difficulties in teaching.)

8. System design exercises:
e.g. design
1. Computer aided Conversation
2. Postal service (or telephone service)
3. Traffic management
4. Information service for a small community
5. Market: buyers and seller systems - from the advertising of needs and availabilities to the implementation of negotiating procedures.

(Dr. Holt feels that in the context of these problems the use of computers would arise naturally and, whether the problems are approached with the help of theory or not, that this type of context study is essential for anyone involved with computers.)

9. The relation of resources to problem definition and solution.

10. The limits of systems thinking as a method.

Dr. Holt said he believed that there were revealable limits to what can be accomplished with explicit definition and systems thinking and that this belief was related to Wittgenstein's statement, which he rendered as "What can be verbally expressed can also be clearly expressed, and what cannot be verbally expressed, about that one should shut up". He also referred to what he feels is one of our culture maladies, namely that everyone is fundamentally uninterested in the limits of anything, though Professor Hamming pointed out Godel's theorem as a famous counter-example.)

11. Representation Seminar: The purpose of this is to develop skill in the construction of representations of formally complicated objects. The skill involves a suitable synthesis in the light of:

(a) The nature of the receiver for whom the representation is constructed.

(b) The range of behaviour toward the object which the representation is to guide.

(c) Representational techniques available (e.g. maps, lists, arrows, colour, text, etc.)

The course would proceed by first choosing an object for representation (together with a receiving context) and the students could then build representations using those techniques they thought appropriate. Then their results would be discussed and criticised, in regard to their success in meeting the specific needs of specific receivers. Examples of such problems are: a traffic pattern (to driver, city engineer); a subroutine (to programmer, to user, to compiler, to keypuncher, to proof-reader etc.); a knot (to the tier, the untier, the classifier etc.); etc.

To finish off, Dr. Holt read two papers he had written which we quote here in full. The first was read in answer to a request from <u>Professor Dijkstra</u> to say some more about the personification of computers; and the second, (which was, he added, written solely for the sake of its last paragraph) as a further elucidation of point 10 above.

The Personification of Computers

That all of us, professionals and laymen alike, personify computers is undeniable. One need only survey arbitrarily selected writings on computer subjects in technical and popular magazines to be convinced of this.[1] The reason is plain. A man interacting with a computer system, or equally, systems interacting with other systems, seems analogous to persons communicating with one another. If, then, one treats computing systems and their parts as "persons", one gains a formidable linguistic advantage: instead of having to invent new words and expressions to describe their structures and functions, one can simply pirate the existing language descriptive of person-to-person communication. True, the semantic setting is new and different (after all, there is, on the face of it, nothing very similar between events in a program execution described as "calling a routine" and mom yelling "Johnny!") but the analogy works.

There is a price for the convenience: the danger of semantic confusion. Naturally enough, it turns out that, although substantial parts of the language of inter-person communication have useful application to computer systems, other parts do not. Yet, having once treated the system entities as persons, one comes to expect that all of the correlative language ought, somehow, to be appropriate and meaningful. From such expectations one can even generate apparent philosophic problems, such as "Can computers think?".

The following are some examples of sentences about persons which are important to the meaning of "person" in our common speech.

A1. Persons communicate with one another.

A2. Persons have goals and interests which may conflict with those of other persons.

A3. Persons have responsibilities which they do or do not fulfill; correspondingly, they may be rewarded or punished; blamed or honored.

A4. By law, persons are accorded rights and protections; also, demands are made upon them.

A5. Persons play roles vis-a-vis one another (friend, enemy, servant, lover, teacher, etc.)

Which of these sentences are well applied to computing systems? Finding the answer depends upon a subtle mix of factual and normative judgement. Nothing about the nature of computing systems forces us to regard them as "persons" at all. C.A. Petri,[2] for example, has proposed a mode of computer system analysis which does not depend upon personification.[3] For many purposes, however, personification is a great conceptual convenience, and for these purposes we should consciously mould the appropriate person concept, rather than being carried forward by the forces of unrecognised motives (such as the need to talk a prospective sponsor into another project on "natural language" communication with computers, or another scheme for "artifical intelligence").

Sentences of all the types A1.A5 (except, perhaps, those involving words like "honored" and "lover") have been seriously used by computer professionals in describing the structure of executive systems, syntax analysers, heuristic programs, etc. Here these sentences have had so clearly delimited contexts that they have not tempted philosophers into deeper speculations about the "true nature" of these routines.

In describing the relations of systems to people, professionals have mostly used sentences related to A1. The lay public has generally carried personification farther (as illustrated in footnote 1). What harm can come from unbridled personfication?

B1. The development of false conceptions about the actual performance capabilities of computer systems.
B2. A detrimental change in the distribution of responsibility and accountability - i.e. the assumption that responsibilities can reside in computer systems, thus losing track of human individuals or groups on whom the responsibilities would otherwise have fallen.
B3. A detrimental change in our appreciation of our own creative functions on the grounds that "computers can do it as well or better".
B4. (A less actual, but nevertheless real possibility.) The introduction upon the social scene of a new fictive class of "interests" and "goals" assumed resident in computer systems and accorded weight and consideration alongside the interests and goals of individuals and groups.

I would like to illustrate B3 with an example based on current system design practice.

There are many extent examples of computer systems which perform some service for "customers" of the organization which installs the system. By a slight extension of the term "customer" one can include computer-aided instruction systems (the students are the "customers") as well as systems for billing, bank accounting, information retrieval, etc. In all of these cases, the customer may be an unhappy one who wishes to complain about the computer-aided service he receives. His complaints may pertain to a particular transaction; to the handling of a series of transactions; to some policy which affects all customers, etc.

Whether the service is computer-aided or not, organizations will normally maintain channels through which complaints are received and adjusted. Installing computer systems is apt to make the existing channels less efficient. ("I'm sorry, lady, but without that number on the upper left hand corner of your yellow slip, there's nothing I can do" ... "Well, you see, our computer reports are produced bi-weekly; until next Monday, there is nothing we can do for you" .. "Yes, I understand, sir; but that's all computer operated now, and I wouldn't know who to direct you to" ... "That's an ill-stated question" - comment of a student to a computer-aided instruction program).

Now the question arises: how about using the newly installed computer to handle complaints? A system designer would probably argue: that's too expensive; it's beyond the state of the art (not enough advancement on artificial intelligence to permit the writing of programs which intelligently answer complaints.) But, that there is a wholly different, and perhaps easily programmable, interpretation of the phrase "letting the computer handle complaints" will probably not have occurred to the system designer. Suppose the system were to be used as follows.

It is given a new class of inputs, called "complaint messages." Depending on the case, these either originate with the customer directly or go through intermediate human encoding. The system's function is to transmit these messages to some proper cognizant person in the organization. Such transmission involves the following computations: (1) one or more addressees of the output message, depending on complaint type; (2) a properly formatted encoding of the complaint; (3) extracts of system file records, giving the pertinent history which bears on the complaint.

The important point is this. Our already unconsciously hardened views of personalized computer systems lead us to think that, if a function is committed to the system, then the system must "do it on its own." In the example, this applies both to the functions about which a customer might complain, and the function of answering complaints.

Complaint message handling as just described could have the effect of making computer system installation the occasion for the improvement of communicative relations among human individuals and groups. We must be careful to fashion our concepts of computer function so that the use of these extraordinary devices will generally tend in this, and not in the opposite, direction.

[1] A London UPI release: "A British mathematician said his computer had decided that Sir Francis Bacon could not have written the plays of William Shakespeare." Time Magazine, April 2, '66: "The computer is, in fact, the largely unsung hero of the thrust into space. Computers checked out all of the Gemini's systems, kept track of the spacecraft's position in the heavens, plotted trajectories, issued commands to the astronauts ..." Page 279 of the Proceedings of the 21st National Conference of the ACM (the page selected wholly at random): "A practical device should be capable of ... correcting for the individual's speech characteristics ... and recognizing the information content of the incoming speech signal."

[2] Carl Adam Petri, Computer Scientist at the Institute of Applied Mathematics, University of Bonn, Bonn, Germany.

[3] A computing system can be viewed as a medium for the transmission of messages between persons. Computing systems as medio have two important new properties (in contrast to other media, such as telephone lines, or the air): (1) the possibility of message arrest in the medium until its further propagation is released by the arrival of another message: (2) the possibility of causing complex and determinate interactions between messages, resulting in their transformation and propagation.

Robotics and Military Applications*

There is an article of faith which underlies the entire idea of Robots to which tasks are delegated under some general specification of goal -- such as "destroy that gun emplacement", "steer this vehicle safely to camp T15", "get me some matches", and the like, namely:

1. That a man, acting with the corresponding intentions, performs a pre-defined computation on a pre-defined class of inputs.

Once this premise is accepted it is then only a matter of working hard, to discover as much as possible about the structure of the inputs and of the computation, so that an approximation to it might be manufactured in the laboratory -- exactly like undertakings in bio-chemical synthesis with the end goal of synthesising a living organism.

2. I do not believe the premise.
3. I think that serious doubt can be cast upon it by methods which responsible scientists would find acceptable.

The basis for my disbelief is not some specific class of existing evidence, but rather my understanding of the relationship between human sense data and the context of human transaction within which sense data arise. My point about this is very simple and yet difficult, so bear with me, if you will.

Consider different occasions on which I might perform an action describable as "turning the car right". Let it be understood that on each of these occasions the action was sensible or ordinary standards -- in other words, appropriate to some understandable context of action. Here are examples of such contexts.

* The thoughts below were stimulated by the discussions of the last day of the ARPA contractors meeting, February 1971, San Diego.

4. I am following a travel plan which calls for a right turn.
5. I respond to an unexpected verbal instruction to "turn right".
6. I want to avoid an accident and therefore deviate to the right.
7. I want to check that the car steering operates properly.
8. I want to crash the car, to avoid some expectable larger disaster.

and etc.

Now on every one of these occasions my action is guided by sense data: however, the significant features which guide my action on these different occasions are different. And now a key question arises:

9. Is it practically possible to regard the sense data as having a fixed structure on all such occasions, a structure which lends itself to the recognition of the various classes of significant features on which decisions in context such as 4-8 depend?

Since I regard the point under discussion as very important, let me try to lend vividness to my question with the help of an analogy.

In different languages, we know, different sets of sensible features are extracted from the stream of speech sound to give stucture and meaning to what is said and heard. Linguists assume that these features are usefully representable by a linear code in which the successive elements represent phonemes. Such representations are, therefore, called phonemic transcriptions of utterances.

From language to language the features of articulation upon which phonemic distinctions depend are different. Thus in one language phonemically different consonants may be distinguished from one another by their degree of aspiration while in another language differences in aspiration may never be used to carry meaning.

Linguists also make use of another transciption of utterances, namely phonetic. Here the idea is to represent, by means of a phonetic alphabet everything which could possibly pertain to distinguishing one utterance from another. In concept this is supposed to be a representation which is independent of subsequent interpretation for meaning -- as unbiased in this regard as, let us say, tape recordings of utterances.

This view of phonetic transcriptions seems to me demonstrably incorrect.

10. Tape recorders can record the sound of a jet engine; phonetic transciptions cannot.

11. If there did not exist languages in which the degree of aspiration is used for phonemic differentation, there would not exist phonetic signs to record such differences.

12. If a language were discovered tomorrow in which the speaker, from time to time, pressed one of his nostrils shut, and the resulting audible differences were used to make necessary distinctions, then the phonetic alphabet would need extension. Note that such an extension would make a post hoc change in the way in which all previously known languages would be phonetically transcribed. (Linguists would then say that the 1-nostril/2-nostril distinction in English occurs in free variation, albeit rarely in its 1-nostril form.)

In short, phonetic transcriptions are not a record of sense data devoid of interpretive intent. (There are no "unprejediced" records.) They are clearly pre-phonemic just as phonemic transcriptions are clearly pre-morphemic. Also, as 12. suggests what is phonetic may be subject to later revision in the light of new assumption about what is phonemic.

We now return to 9. Answering 9 in the positive would mean expressing one's belief in the following:

13. The existence of a counterpart to the phonetic alphabet for the representation of sense data relevant to the steering of a vehicle competently around an airport.

14. The existence of "pattern recognition" algorithms which, with respect to this "universal" sense data, are able to extract the significant features on different occasions, such as those in 4-8. This seems to be related to but not identical with an analogous demand that, given a target language and samples of phonetic transcriptions of its utterances, one be able to find, in these transcriptions, the appropriate phonemic boundaries.

In a less abstract vein, consider the following sample question:

15. What relation is there between the significant features of an object which I wish to avoid hitting, and those of an object steering a vehicle one might want to do both.)

Anyway, 13 and 14 seem to me the subjects of doubt for even a relatively simple domain of practical operation, such as steering a vehicle from one place to another.

I believe that the central question which I am raising is absolutely unaddressed by current work in Robotics because the logical structure of the goals as defined to experimental robots are not practical goals within my understanding of the term 'practical'. I believe that our understanding of the simplest of instructions in practical situations - that is, not situations which have been a priori isolated from the rest of the context of daily life by being labelled "an experiment" -- there enters a large, unexplicit, and in principle unexplicable context of human interests which govern our response to the instruction, both in its sensing and effecting aspects. If someone says to me "get me some matches", how much and what kinds of effort will I make in response? At what expense? Well, ... that depends. On what? I cannot say.