THE LOGIC OF PROTECTION

L. Kohout[*][†]
[†]University College Hospital
Medical School,
University of London, U.K.

B.R. Gaines[*]
[*]Man-Machine Systems Laboratory,
Dept. of Electrical Engineering Science,
University of Essex, Colchester, U.K.

Abstract This paper presents a brief exposition of the role of various mathematical techniques in the development and utilization of resource protection structures for computers. The first section is concerned with the semantics of the problem - the distinction between protection problems in general and those whose complexity necessitates deeper theoretical treatment. The second section considers the roles of algebraic, topological, and modal/multi-valued logic, techniques in the analysis of protection. Finally we give an analysis of a current protection model to illustrate the problems and techniques.

1. The Problem of Protection

1.1 Introduction

The protection of the security of potentially shared resources, both information and activities, has become a problem of major interest in computer science and engineering. Fundamentally the problem is not different from those of personal, commercial and government security in the pre-computer era - the differences are quantitative ones of monitoring electronic activities whose speed, magnitude and inaccessibility far exceed the human transactions they mimic. Technically, aspects of security peculiar to computer-based systems may be seen to arise with the early time-sharing systems such as CTSS and MAC [1] which broke away from batch-processing of naturally isolated jobs and allowed users to share not only basic resources like storage and processing power but also to access joint data bases and processes for mutual interaction in real time. It was the announcement of the MULTICS [2] project in 1964, particularly the discussion of its aims and objectives in a group of 6 papers at the 1965 FJCC, that awoke the computer community at large to the new technical problems, as well as the new potentialities, of systems accessed simultaneously by multiple, competing and collaborating, users. Even at this early stage the social implications of such systems were discussed [3] and these have become a matter of increasing public concern in recent years [4,5].

Thus protection has arisen as an important and distinct problem in its own right. It is closely associated with many of the technical problems of operating systems, e.g. ensuring the correct functioning of co-operating sequential processes [6], but these may be seen as prerequisites to the implementation of protection rather than central to the problem itself. Equally the availability of adequate protection structures is itself a prerequisite to the full exploitation of techniques of modular [7] or structured [8] programming. Perhaps the neatest way to make the distinction is to note that the natural logics of protection are not the Boolean algebras so basic to computers, but rather the modal logics [9,10] of possibility and necessity (alethic), permission and obligation (deontic) - for example, we typically wish to know whether it is possible for a process which is permitted to access a data structure, but obliged to obey certain synchronization disciplines in changing it, to avoid these, or whether they are necessarily obeyed (hardware enforced). Ensuring that the disciplines are available (e.g. through semaphore mechanisms) and using them to ensure a formal and enforceable match to the problem structure (e.g. a hierarchy of processes) are not strictly part of the problem of protection itself. The central problem is that of the logic of protection, its consistency and its implications in particular implementations. This would be a comparatively straightforward problem were it not for the extremely dynamic nature of the environment in which the logic operates.

1.2 Motivation and Structure of Paper

This paper presents a brief exposition of the role of various mathematical techniques in the development and utilization of resource protection structures for computers. On the one hand we are concerned to present the problem as a new systems area, similar in status to such areas as identification, stability and control, and worthy of the attention of theorists. On the other hand we are concerned to investigate the nature and magnitude of practical requirements for, the current implementations of, protection structures to ensure that theoretical developments have a proper and useful semantics.

The studies reported arose from our experience in the design of a descriptor-organized minicomputer [11,12] in which the full power of hardware-enforced ring crossing processes may be invoked by procedure calls in high-level languages [13]. We became aware of the potential for essentially simple protection mechanisms to lead to complex dynamic problems that defied human intuition, and were led to investigate the applicability of logical [14] and topological [15] models of the phenomena involved. We found there to be a conflict between the essential simplicity of use of protection mechanisms in most current systems, and the theoretical complexity that could arise. The resolution is probably that the use of the capabilities of computers to administrate large organizations in a totally integrated fashion [16] is rare as yet. Most users of computer utilities still use them for economic reasons only and require a null-relationship of total confinement with other users.

The place of more complex analyses of protection is discussed in the following section. The middle section is concerned with the interplay between algebraic, topological and logical techniques in this problem area, and serves to introduce the final section which presents an example of their relative roles in relation to a model of protection based on that of Graham and Denning [17]. It is inappropriate in this paper to attempt to survey the many contributions to the protection literature, and we refer the reader to truly excellent recent survey of Popek [18] which lists some 84 references.

2. The Semantics of Protection Structures

2.1 Is There a Problem ?

Before any theorist moves in with an armoury of mathematical techniques it behoves him to ensure that the enemy actually exists and that he is not finally solely engaged in grappling with his own terminological obscurity. Any computer manager will confirm that his installation has a security problem. However his anecdotal reports are more likely to demonstrate human errors, software bugs and design faults, rather than any deep and elaborate failures. His problem is still security in the negative sense of containment, and the hardware mechanisms of most commonly used machines are designed with this in mind.

Even MULTICS, with its objectives of supporting collaborative user communities, is based on a simple linear order of protection rings of monotonically decreasing capability which it is simple to express logically. It allows users to share, or not to share, major data objects but does not realistically support more subtle interactions between them. The wide use of computer systems with far less complex protection facilities than MULTICS is evidence that a substantial part of the user community can get by without such subtlety for their current activities. This does not prevent them being adversely affected when manufacturers attempt to incorporate it, unsuccessfully, in their operating systems, but it indicates that we have to search with care for the positive requirements.

2.2 Capabilities and the Graham and Denning Model

A key paper in expressing these positive requirements and mechanisms for their satisfaction is that by Lampson [19] who introduces the term capability for the access right that a process may possess to an object, a generalized resource. Capabilities are themselves protected objects which may be created and passed between objects only according to prescribed rules. Graham and Denning [17] make explicit appropriate rules for the manipulation of capabilities in a second key paper. It is important to note that although these papers have abstracted the protection problem with a high degree of generality, the exemplary semantics given is still very simple (in terms of capabilities to read and write into files) and many basic problems are deliberately excluded (for example, access to data being dependent on its value). The concluding paragraph of [17, p.428] is particularly important in summarizing the state of the art.

Hardware realizations of capability-based protection structures are being developed [20] and at least one commercial machine is now in production [21]. The Graham and Denning model clearly merits investigation and extension in its own right [22]. However, the semantics provided by current protection hardware and even advanced operating systems is probably inadequate to justify such an investigation and certainly inadequate to assess the results. We can find a far richer semantics in the problems of large data-bases and information systems.

2.3 Data-Bases and Data-Interrupts

A key paper on data-base protection is that by Conway, Maxwell and Morgan [23] who consider security requirements in practical information systems such as personnel records. Here the units which must be protected are far smaller than those previously considered, being individual fields in a single record rather than complete files of information. Equally importantly the rights to access certain fields may be dependent on the data stored in these or other fields of the record. Thus a typical protection predicate might be: "an assistant manager may read the personnel records except medical history of employees in his division with salaries of less than $30,000". This level of detail coupled with the size of the data-base provides far richer and more complex examples of protection predicates than does that on operating systems.

What these examples lack, however, is the dynamic complexity of operating systems in which the protected objects are not only passive data items but also active processes which themselves initiate further activities and accesses to protected items. This may be introduced into the data-base problem by considering a suggestion of Morgan [24] of "an interrupt based organisation for management information systems" in which a predicate on the values of data items may be used to invoke a process. For example, an inventory control system might have processes attached to variables indicating stock levels that automatically re-order items if the stock falls below a prescribed level. Zelkowitz [25] has suggested a hardware implementation of this mechanism on the IBM360 and it is feasible with any tagged [26] or descriptor-based [11] machine in which the tags are retained in file structures.

Examples of data-interrupts in use are currently probably found only in such "artificial intelligence" languages as CONNIVER [27]. However, the use of "data-base-driven" processes is very much in line with concepts of modular programming [7] since they allow an activity dependent upon the value of a variable to be implemented as a single independent module rather than incorporated as conditional calls in every routine that may update that variable. They have a natural place in languages such as POP2 [28] and EL1 [29] which allow an "updater", or type-coercion routine, to be associated with an individual variable. Their availability is particularly attractive in quite simple transaction-processing systems where on-line users access the same data-base, e.g. dealing systems [30], since all activities naturally centre around, and are driven by, the state of the data-base. Whilst the hardware necessary to implement the data-interrupt is comparatively new, we have reported elsewhere [31] the practical success of commercial and medical transaction-processing systems based on the interpretation of a high-level language on a minicomputer, and are currently extending the facilities to include data-interrupts, a simple extension to an interpretive language.

2.4 Summary

Thus a combination of the finely detailed, data-dependent protection requirements of data-base systems together with the dynamic protection requirements of data-interrupt driven systems provides a far richer semantics for models of protection than does the conventional "operating-system" requirements, and one that is both generated by current needs and is feasible in many applications with current hardware/software technology. The potential of such systems is well beyond our current intuitive conceptions of what computer systems can do. The possibility of adding arbitrary distinct processes, "unknown" to one another but mutually interacting through changes in state of a common data base, allows a far more natural development of a system, based on mimicing the activities of individuals in an organisation. Equally such a system may grow rapidly beyond the comprehension of its designers since the addition of a new activity may invoke a host of natural side-effects which have no referents whatsoever in the new activity itself. The problem of ensuring adequate security whilst at the same time taking full advantage of the mutual collaboration possible will become acute.

3. The Mathematics of Protection

3.1 The Roles of Different Formal Models

The natural representation of a protection structure relating processes to capabilities, adopted for example in both our key references [17,23], is that of a *matrix* expressing the (algebraic) relation between them. Such relations, expressed as matrices, can also model the dynamics of protection, the permission to pass a capability to another process, etc. The overall model obtained is naturally *automata-theoretic* with its analytic basis being clearly *algebraic*.

The algebraic model itself has a direct application to questions about procedures to follow in attaining certain *aims*. "How do I write into file A", is answered by enumerating trajectories of communication through processes which do not violate the protection. There may be none (not allowed), a unique solution or many possibilities with different properties. This corresponds to a control problem in the state space of the protection automaton.

However, many of the major questions of security are not of this nature but relate more to global properties of reachability, "can any of these processes *access* this information", "is this process *contained* in this domain". Such questions are naturally ones of *closure* [15] and best treated within a topological framework. They may be seen as *stability* problems in the state space of the protection *automaton*.

The actual closure spaces generated by any particular protection structure should reflect the intentions of users in setting it up. There are direct formal relations between such spaces and modal logics [32,33] so that the semantics of the model may be expressed in a communicable form. It is easier to understand, "it is desirable to do X and it is permissible to do Y but the system will not allow you to do Z", or, more globally, "the protection system of the HCN471 will not allow the user to follow this desirable practice and is dependent upon him obeying these rules", rather than "$X \varepsilon S_a(U)$, $Z \varepsilon S - S_c(U)$", or, "the HCN471 has no compatible closure relation".

In practice although both topological and modal logic techniques and vocabularies are useful, any real protection structure will be finite and users will tend to superimpose on it a readily understood structure of nested protection domains. The many-valued logics thus generated may be formally regarded as finite approximations to modal logics [32], and are an alternative natural expression of hierarchical, ordered structures (e.g. protection rings).

From a category-theoretic point of view [34,35,36] these distinctions are purely ones of terminology and perhaps the ultimate abstraction of protection structures should be expressed categorically. However, although the old lines of demarcation no longer exist, the old terminologies are still evocative and what is clumsily expressed in one may become quite elegant and transparent in another. Thus, in summary, we see the appropriate use of mathematical tools in the study of protection to be:

Algebraic formulation of protection axioms → topological formulation of closure properties → modal logics of resultant spaces → multi-valued logic representation in finite matrices.

3.2 The Graham and Denning Model

As noted in section 2.2 the best developed formal model of protection is that presented in [17], and we have based our analysis in the following section upon this. Briefly, Graham and Denning distinguished "subjects" which are active entities (a process and domain of access to resources) and "objects" which are essentially resources to which access must be controlled - a "subject" is also an "object". They represent a protection structure as a matrix of subjects against objects giving the access rights of each subject to the objects (including other subjects), together with a set of rules for changing the matrix (e.g. by adding or deleting subjects and objects).

The elements in the matrix form "capabilities" (an access right by a subject to an object) and the dynamics of the model arise to a large extent because capabilities can be passed from subject to subject. It is possible to treat the right to pass a capability (the "copy flag" in [17]) itself as a capability and such generality is desirable for theoretical compactness. However, in explaining the model it is useful to separate out the protection matrix from its dynamics and we introduce a pass as the right to pass a capability, and a permit as the right to give this right - further recursive extension is unnecessary to the example.

One extension we have not made in our analysis is to consider relationships and interactions between capabilities. In management information systems it is unlikely that the capabilities would be themselves simple, unitary actions. Rather they would reflect the fine structure of possible actions so that a major action, such as writing into a record, would be possible only to the possessor of multiple capabilities. Equally the act of so doing is likely to be necessarily accompanied by other acts, e.g. associated with transaction monitoring. This implies that there will be rather more complex relationship between capabilities and actions than is assumed in any current model, but the extension to allow for this is straightforward. The only remark we make for the moment is that the algebraic structure of interaction between capabilities must be positive (in the sense of [37, p.125]), i.e. one capability cannot cancel another out. This is implicit in the literature, but it is tempting in extending the models to add "anti-capabilities" (for example to allow a user of a subsystem to ensure that it is "memoryless" by removing its access to certain channels of communication). Non-positive capabilities make nonsense of the use of closures, and do not seem to have a proper place in the semantics of protection.

Two further concepts are necessary which are relevant to the use of Graham and Dennings model rather than its structure. Some ("privileged") subjects will have capabilities that would show up as dangerous in any analysis but which they will not use. We introduce an intention matrix that specifies what ones will be used. This enables the closures computed to reflect relationships of trust between subjects. In analysing his protection a user would adjust an intention matrix to specify his own use of capabilities (assuming other users have malicious intentions) and a trust matrix to prevent non-significant paths for protection failure being continually drawn to his attention, but both may be represented in the model as a single matrix.

4. One Formal Model of Protection

4.1 A Concrete Example

The terminology of the following sections would be opaque without some concrete examples. Unfortunately examples tend to be either trivial or too lengthy in description. The following artificial situation has been generated to serve as a basis for illustrating each technique discussed.

Start of example *The company X runs a network of data processing systems. The basic flow of information is shown in Fig. 1: the system x can directly inspect $x1$ and $x2$, and indirectly inspect $x3$ and $x4$ or $x5$ via $x1$ and $x2$, respectively. In addition to this fixed hierarchical flow, the systems can exchange information within the network according to certain dynamic relations.*

The type of problem we shall study is that there is exchange of information with similar systems operated by competitors: $x5$ with $y5$ of company Y and $x4$ with $z5$ of company Z. Y and Z must not obtain the information in x, $x1$ or $x2$ at the same time, although each part of the information on its own, or combinations at different times (say more than t a part) are harmless. The information flow is fully defined by a sequence of action, pass and permission relations. Computationally these might be represented as (sparse) matrices but for this text we shall work with the relations.

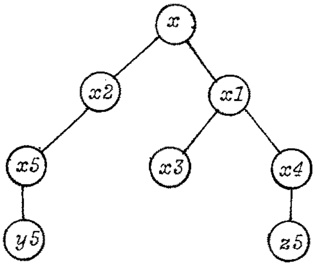

Figure 1 Data-Processing Network

4.2 Terminology and Definitions

In our terminology, we shall stress the dynamical character of protection.

Participants - abstract elements of a protection structure, which can be either subjects or objects. The set of all participants will be denoted by $X = \{x_1, x_2, \ldots\ldots, x_n\}$.

An object - a participant, manipulation of which must be controlled.

Subject - an active participant whose manipulation of objects must be controlled.

A participant x_j can simultaneously be a subject with respect to the object x_i and an object with respect to the subject x_k.

Action - certain precisely specified behaviour of participants. A subject acts on an object, and an object is manipulated by a subject. (Examples of action: read, write, seek, execute, etc.).

Activity - a sequence of actions with some unambiguously specified purpose.

Aim - an a priori specified (required) result of a sequence of actions, which form a particular activity. Note that a specific action can enter as a component into the formation of two or several distinct activities.

Aim controllable by a group of subjects X_t - an aim which can be achieved by a sequence of actions exclusively performed by the group X_t.

Aim protectable by a group of subjects X_t - an aim which cannot be achieved by an activity outside X_t without the specific permission of the group X_t.

It is important to realise that a certain specific action can form two or more distinct activities, or can contribute to the fulfilling of two distinct aims. Hence there may exist two different and often contradictory requirements of the protection in a case where the same action is a component of two distinct activities.

Action matrix - for an action α_i is defined by a relation $R_{\alpha_i}(x_j,x_k)$ between participants from $\{X\}$.

Capability - a protected name, a pair $\langle\alpha_i,x_j\rangle$ where α_i is an action and x_j is an object. A subject x_k has the capability $\langle\alpha_i,x_j\rangle$ if it can perform the action α_i on the object x_j.

A subject can pass a capability it holds to another subject. This action must be properly controlled. For this purpose we shall introduce

Pass - a protected name, a pair $\langle\langle\alpha_i,x_j\rangle ,x_k\rangle$ where $\langle\alpha_i,x_j\rangle$ specifies the capability and x_k is the subject holding the pass. A pass signifies that a subject x_k is allowed to pass a capability.

Permit - a protected name, a pair, $\langle\langle\alpha_i,x_j\rangle ,x_k\rangle$ where $\langle\alpha_i,x_j\rangle$ specifies the capability to which the pass refers and x_k is the subject which holds the permit; $\langle\langle\alpha_i,x_j\rangle ,x_k\rangle$ signifies that the subject x_k can give the permission to pass the capability $\langle\alpha_i,x_j\rangle$.

4.3 Algebraic Models

An abstract algebraic model used for the investigation of the dynamics of protection structures, is formed by relations expressing the mutual dependencies of subjects and objects as well as relationships of capabilities, passes and permits.

The set A of all actions α_i, which are elements of an activity Z_k is denoted by:

$$A(Z_k) = \{\alpha_1,\alpha_2, \ldots\ldots,\alpha_\omega\}$$

The structure of an action α_1 can be described by the triple of relations $R_{\alpha_i}, R_{\phi_i}, R_{\pi_i}$:

$$R_{\alpha_i} = R_{\alpha_i}(x_k,x_l) \qquad R_{\phi_i} = R_{\phi_i}(x_k,x_j,x_m) \qquad R_{\pi_i} = R_{\pi_i}(x_k,x_j,x_m)$$

The relation R_{α_i} defines the subject-object relationship and specifies the capabilities of a set of subjects $\{sub\} \subset \{X\}$ to perform the action α_i on the set of objects $\{ob\} \subset \{X\}$.

The ternary relation R_{ϕ_i} specifies which subject x_k can pass the capability $\langle\alpha_i,x_m\rangle$, $x_m \varepsilon \{ob\}$ to a subject x_j. The ternary relation R_{π_i} specifies which subject x_k can give permission to copy the pass $\langle\langle\alpha_i,x_m\rangle ,x_j\rangle$, $x_m \varepsilon \{ob\}$, $x_j \varepsilon \{sub\}$.

Each ternary relation R_{ϕ_i} , R_{π_i} can be expressed as a set of binary relations:

$$R_{\phi_i}(sub_1, sub_2, ob_m) \equiv \{R_{\phi_i,ob_1}(sub_1, sub_2), R_{\phi_i,ob_2}(sub_1, sub_2), \ldots$$
$$\ldots, R_{\phi_i,ob_\alpha}(sub_1, sub_2)\}$$

where $m = 1,2,3,\ldots\alpha$; sub_1, sub_2, $ob_m \varepsilon \{X\}$. Similar expressions hold for R_{π_i}.

The relations which have been so far described deal with permissions. However, it seems necessary to introduce structures which can describe the intentions of the participants, as well as the permissions. This can be exemplified by the following example. Let us consider the permission which is described by the transfer rule R1 of Graham and Denning. The rule R1 permits a subject to transfer any capability it holds to any other subject, provided the donor has the corresponding pass (which is realised in the scheme as a copy flag). Without the introduction of some further structures we can investigate only the case where the intention of each subject with the appropriate pass is to give capabilities to all subjects. This limit case describes only the minimal restrictions which are enforced by the permission rules but not the actual state of the protection system in the case that the participants do not reach the limits forced by the permission rules.

However, this is required by a user who would like to find out how he should pass his capabilities and avoid some unwanted side effects.

Now we shall introduce a formal definition of a model of protection structures. It will be shown later (section 5) that the model can be interpreted as a hierarchy of sequential machines.

Definition

A model $\mathcal{M}(Z_k)$ of an activity Z_k is composed of the set of triples:

$$\mathcal{M}(Z_k) = \left\langle R_{A(Z_k)}, \Phi_{Z_k}, \Psi_{Z_k} \right\rangle = \left\{ \left\langle R_{\alpha_i}, \Phi_i, \Psi_i \right\rangle \right\}_{i=0}^{i=\omega}$$

where α_i runs over the set $A(Z_k)$ of all actions, which are the elements of the activity Z_k; i.e. $A(Z_k) = \{\alpha_1, \alpha_2, \alpha_3, \ldots\ldots, \alpha_\omega\}$.

$\Phi_i = \left\langle R_{\phi_i}, R_{\pi_i} \right\rangle$ belongs to the permission structure Φ_{Z_k}.

$\Psi_i = \left\langle R_{\gamma_i}, R_{\sigma_i} \right\rangle$ belongs to the intention structure.

The relation R_{γ_i} defines the interrelations between the intended passes, and R_{σ_i} between the intended permits in a way which is analogical to the definitions for the permission structure Φ_i. The difference between Φ_i and Ψ_i is only in the semantics.

In general, changes in the structure can be made by actions ϕ_i which operate on Φ_{Z_k} and which change the $R_{A(Z_k)}$ or by actions ψ_i which operate on Ψ_{Z_k} and change Φ_{Z_k}.

A trajectory of $\mathcal{M}(Z_k)$ is an admissible sequence of actions $\alpha_i \alpha_j \psi_i \phi_r \alpha_k \ldots\ldots \alpha_j \psi_i \phi_r \ldots\ldots \alpha_k \ldots\ldots$.

The dynamics of a participant is the current state of the vector

$$\mathrm{Dyn}_{Z_k}(x_k) = \left\{ \left\langle \alpha_i(x_k), \phi_i(x_k), \pi_i(x_k), \gamma_i(x_k), \sigma_i(x_k) \right\rangle \right\}_{i=1}^{i=\omega}$$

Only certain sequences of actions are admissible. The admissibility of sequences must be specified by some additional rules which depend on the type of activity and on the character of actions.

Example Continued - *the set of all actions* $A(Z_k) = \{\alpha_1, \alpha_2, \phi_1, \pi_1\}$

α_1 ..... *inspect data* $\qquad \phi_1$ ..... *pass the capability* *inspect* *data*

α_2 ..... *record data* $\qquad \pi_1$ ..... *permit to pass the capability* *inspect* *data*

capabilities are defined by the action relations:

$$R_{\alpha_1} = \{(x,x),\ (x,x1),\ (x,x2),\ (x1,x1),\ (x1,x3),\ (x1,x4),\ (x2,x2),\ (x2,x5),\ (x3,x3),\ (x4,x4),\ (x5,x3),\ (x5,x5)\}$$

$$R_{\alpha_2} = \{(x2,x2),\ (x3,x3),\ (x3,x5),\ (x5,x5)\}$$

passes are defined by $\{R_{\phi_{1,x}}, R_{\phi_{1,x1}}, R_{\phi_{1,x2}}\} \equiv R_{\phi_1}$ *where*

$R_{\phi_{1,x}} = \{(x,x1)\};\ R_{\phi_{1,x1}} = \{(x,x3)\};\ R_{\phi_{1,x2}} = \{(x,x5)\}$

permit is defined by $R_{\pi_{1,x1}} = \{(x3,x4)\}$

model of an activity $\mathcal{M}(Z_k) = \{R_{\alpha_1}, R_{\alpha_2}, \Phi_1, \Psi_1\}$ *where*

$$\Phi_1 = \{\langle R_{\alpha_{1,x}}, R_{\alpha_{1,x1}}, R_{\alpha_{1,x2}} \rangle, R_{\pi_{1,x1}}\} \equiv \{R\Phi_1, R\Pi_1\}$$

$\Psi_1 = \mathbb{1}$ *(universal relation , i.e. every element is in relation to all others)*

The intention structure in this example is the universal relation, which means that the intention of the participants is to go to the limits which are permitted by the permission structure. (Note that only the passes and permits which are related in the permission as well as in the intention structure can be used - the disjunction of the structures).

The trajectory $\alpha_1\alpha_2\phi_1$ *is the sequence of the following actions:*

(inspect) (record) (modify the R_{α_1} *according to the pass relation* $R\Psi$*)*

Let us choose the initial dynamics of the participant x1

$Dyn_{Z_k}(x1) = \{\ \alpha_1(x1), \Phi_1(x1), \pi_1(x1)\ \}$ *where the ranges of the relations are*

$\alpha_1(x1) = \{x1,x3,x4\}\ ;\ \ \phi_1(x1) = \{(x,x1),(x,x3)\}\ ;\ \ \pi_1(x1) = \{(x3,x4)\}\ ;$

If the action ϕ_1 *is applied, it causes the following changes:*

$\alpha_1(x1) = \{x,x1,x3,x4\}$

Now, if the action π_1 *is applied, then* $\phi_1(x) = \{(x,x1),(x,x3),(x3,x4)\}$.

4.5 Rules for Composition of Actions

Rules for composition of actions entering into an activity Z_k cannot be entirely arbitrary. The set of admissible sequences of actions is determined by the type of activity and by the objectives of protection. However, it should be noticed, that the rules of composition also depend on the characteristics of a protected system. Let us take as an example the action 'read'. The previously quoted statement of Graham and Denning " ... reading implies ... the ability to read and copy file ..." means that in the system they had in mind the capability 'read' is equal to the capability 'read/write' in certain activities. We can, of course, design a monitor which would allow us to introduce the capability 'read' without the above mentioned unwanted consequences. From this example we can make some fairly general conclusions, which have impact not only on the design of protection structures as such, but what is more important, on the design of the whole system. That is, elementary actions should be chosen in such a way as to limit the consequences of uncontrollable transitivity of actions.

Now we shall introduce an appropriate semantics into our model in order to be able to handle this problem. An action of one participant upon another is called a direct action if there is no other participant involved as a mediator. An indirect action is an action in which a participant achieves certain aims with respect to another participant through a third participant or through a chain of participants.

Let x_i perform an action α_k on x_j, defined by $R_{\alpha_k}(x_i,x_j)$. We shall abbreviate this by $(x_i \xrightarrow{\alpha_k} x_j)$. Then we can give the following reduction rules, where the symbol o means the composition of actions:

$$\frac{(x_i \xrightarrow{\alpha_r} x_j) \text{ o } (x_j \xrightarrow{\alpha_r} x_k)}{(x_i \xrightarrow{\alpha_r'} x_k)}$$

a transitive action which composed gives an indirect action α_r'; note that the direct action $(x_i \xrightarrow{\alpha_r} x_k)$ is not always defined.

$$\frac{(x_i \xrightarrow{\alpha_r} x_j) \text{ o } (x_j \xrightarrow{\alpha_r} x_k)}{(x_j \xrightarrow{\alpha_r} x_j) \vee (x_j \xrightarrow{\alpha_r} x_j)}$$

an intransitive action either $(x_i \xrightarrow{\alpha_r} x_j)$ or $(x_j \xrightarrow{\alpha_k} x_k)$ or both

More generally:

$$\frac{(x_i \xrightarrow{\alpha_r} x_j) \text{ o } (x_j \xrightarrow{\alpha_s} x_k)}{(x_i \xrightarrow{\alpha_p} x_k)} \qquad \frac{(x_i \xrightarrow{\alpha_r} x_j) \text{ o } (x_j \xrightarrow{\alpha_s} x_k)}{(x_i \xrightarrow{\alpha_r} x_j) \vee (x_j \xrightarrow{\alpha_s} x_k)}$$

Again similar rules can be given for passes and permits.

The action α_1 (inspect data) is not transitive and a corresponding indirect action cannot be formed by a simple composition of two direct actions α_1. For example, taking the subjects x,x2,x5, we get:

$$\frac{(x \xrightarrow{\alpha_1} x2) \text{ o } (x2 \xrightarrow{\alpha_1} x5)}{(x \xrightarrow{\alpha_1} x2) \vee (x2 \xrightarrow{\alpha_1} x5)}$$

The action α_2 (record data) has different properties. For example, if x3 *records data into x5, and x5 into x2 consequently, then x2 owns the data of x3 although x3 cannot write into x2. This is an example of the indirect action* α_2' *. Take the participants x2,x3,x5 and look at the reduction rules:*

$$\frac{(x3 \xrightarrow{\alpha_2} x5) \circ (x5 \xrightarrow{\alpha_2} x2)}{(x3 \xrightarrow{\alpha_2} x2)}$$

The indirect action α_1' *(inspect data of ...) can be formed by the composition of* α_1 *and* α_2*. For example, if x3 records its information into x5 and x2 inspects x5, then x2 is able to inspect indirectly x3. Let us look at some interesting cases:*

for the activity $\alpha_2\alpha_1$ *we get:*

$$\frac{(x3 \xrightarrow{\alpha_2} x5) \circ (x2 \xrightarrow{\alpha_1} x5)}{(x2 \xrightarrow{\alpha_1'} x3)} \qquad \textit{(indirect } \alpha_1'\textit{)}$$

but for the activity $\alpha_1\alpha_2$*:*

$$\frac{(x2 \xrightarrow{\alpha_1} x5) \circ (x3 \xrightarrow{\alpha_2} x5)}{(x2 \xrightarrow{\alpha_1} x5) \vee (x3 \xrightarrow{\alpha_2} x5)} \qquad \textit{(no indirect action)}$$

Following is the result of the activity $\alpha_1\alpha_2\alpha_1$*:*

$$\frac{(x5 \xrightarrow{\alpha_1} x3) \circ (x5 \xrightarrow{\alpha_2} x5) \circ (x2 \xrightarrow{\alpha_1} x5)}{(x2 \xrightarrow{\alpha_1'} x3)} \qquad \textit{(indirect action)}$$

5. Hierarchical Structure of the Protection Model and its Description by Systems of Logic and Topology

The crucial feature of the model $\mathcal{M}(Z_k)$ is the highly specific hierarchical interrelation of its composing structures which forms a hierarchy of sequential machines. This static hierarchical structure as well as the dynamics of the model can be expressed in modal or many-valued logics or by general topological structures which can be made mutually interchangeable. It is necessary to distinguish three qualitatively different actions in the sequence of admissible actions: firstly, actions of subjects on objects, as they are enabled by capabilities, secondly, actions of subjects on other subjects which amount to the passing of capabilities, and thirdly, actions of subjects on other subjects which permit the transfer of passes. Hence, three qualitatively distinct levels appear in the dynamics of the whole model, as well as in the dynamics of the individual participants. This becomes obvious if the last statement is re-interpreted in terms of abstract automata.

The relation between subjects and objects which is described by the R_α of the model, represents in these terms a finite-state automaton, acceptor, which accepts all admissible sequences of α-actions. The set of all participants represents states and the transitions are represented by individual actions on participants. Similar finite-automata describe the R_ϕ and R_γ components of the model (passes). If the R_ϕ and R_γ both accept an action, which means the passing of a capability, the structure of the R_α will be modified i.e. a new transition added into the R_α automaton. At the same time, if the automata corresponding to R_π and R_σ accept the same action, the permitted passes and intended passes will be modified (i.e.) new transition added into R_ϕ and R_γ automata respectively.

5.1 Topological Models

As we stated above (section 3.1), questions about behaviour of participants and about possible violations of protection can be formulated in terms of reachability and controllability in the state-space of a protection automaton. Reachability and controllability can be discussed in terms of generalised closures in extended topologies [15] which have been shown to be semantic models of some modal logics [32], [33]. The considerable advantage of the topological approach consists in the fact that the topological structure 'forgets' parts of the automata structure which are inessential to the dynamics of the behaviour of participants. We can look at the behaviour either of mutually suspicious groups of processes, or of several rival groups inside which the member participants cooperate etc.

We shall use some elements of the theory of generalised (extended) topology in the sequel, the basic definitions of which are given in [15] together with more details and an extensive annotated bibliography on the subject.

Closures in generalised topologies offer a tool for investigation of the dynamics of protection as well as of its limit case established for infinite strings of admissible actions.

The basic element of the topological model is the direct action (pass, permit) closure $\alpha_i(f_i,g_i,r_i,s_i)$ generated by the action α_i. It is defined as a mapping on the power set of all participants:

$$\alpha_i : \mathcal{P}(X) \to \mathcal{P}(X) \qquad \alpha_i(A) = \mathop{\mathcal{E}}_{x_j \varepsilon A} \alpha_i(x_j) = \mathop{\mathcal{E}}_{\alpha_i}(A) \qquad A \subset \{X\}$$

It represents the set of all participants (objects in this case) which can be acted on by the subset A of the set of all participants by a direct action α_i in a particular activity.

An important closure derived from the direct closure action closure is the AIOU-modification [15] of the given A-topology. For this (transitive) closure the important U-axiom $u(u(x_i)) = u(x_i)$ holds. In terms of control and automata theory it is the <u>region of reachability</u> i.e. the limit case of propagation of the effect of particular action or a set of actions. In modal terms, it defines the <u>possibility</u> of the existence of the effect of a selected action on the participants which are members of that closure.

Propagation of the effect of a set of actions which is given by a particular trajectory of $\mathcal{M}(Z_k)$ (i.e. by a selected admissible sequence of actions) can be investigated using iterations of the above defined closures. The k-th iteration will be given by

$$c_i^k(A) = c_i(c_i^{k-1}(A)) \qquad c_i \varepsilon (a_i, f_i, g_i, r_i, s_i); \qquad A \subset \{X\}$$

<u>Example Continued</u>

We shall examine the direct closure a for the action α_1. We have already shown that the action α_1 is not transitive. Hence, the closure a will also determine the limit case of propagation of a_1 action. We shall list the closures of all *singletons of the example*

$a(x) = \{x,x1,x2\}$ $a(x2) = \{x2,x5\}$ $a(x4) = \{x4\}$

$a(x1) = \{x1,x3,x4\}$ $a(x3) = \{x3\}$ $a(x5) = \{x3,x5\}$

$a(X)$ is the set of all participants whose files can be inspected by X. Now, let the trajectory of the system be $\alpha_1\alpha_1\alpha_1$ .... $\alpha_1\phi_1\pi_1$, that is the pass is presented and a permit is given in this sequence. This will cause the following change in the closures from above:

$a(x1) = \{x,x1,x2,x4\}$ $a(x4) = \{x1,x4\}$

$a(x3) = \{x1,x3\}$ $a(x5) = \{x1,x3,x5\}$

Let us designate $\alpha = \alpha_1\alpha_1\alpha_1 \ldots\ldots \alpha_1\phi_1\pi_1$ and $\alpha_3 = \alpha_1\alpha_2$ then for the trajectory $\alpha\alpha_3$ we shall list the closures for all participants:

$a(x) = \{x,x1,x2\}$ $a(x2) = \{x2,x5\}$ $a(x4) = \{x1,x4\}$

$a(x1) = \{x,x1,x3,x4\}$ $a(x3) = \{x1,x3\}$ $a(x5) = \{x2,x3,x5\}$

The effect of the new application of the action $_3$ (i.e. the resulting trajectory $\alpha\alpha_3\alpha_3$) is computed by the second iteration $a(a(x_i)) = a^2(x_i)$. It will change the following closures:

$a^2(x2) = \{x,x1,x2,x3,x5\}$ $a^2(x3) = \{x,x1,x2,x3\}$ $a^2(x\ ) = \{x,x1,x2,x5\}$

For the third iteration (the trajectory $\alpha\alpha_3\alpha_3\alpha_3$) we get the changes:

$a^3(x) = \{x,x1,x2,x3,x5\} = a^3(x5) = a^3(x2)$ $a^3(x3) = \{x1,x3\}$

$a^3(x1) = \{x,x1,x3,x4\}$ $a^3(x3) = \{x1,x3\}$

Further iterations (applying α_3) will not change the closure. We can see that we have computed the transitive closure (the U-modification of the original topology). This determines the worst case of the security in the system.

From this last computation it can be seen that the requirement on the security specified in the above has been violated so that the competitors can obtain the content of the data files of $x,x1,x2$ from $x5$ at once. Hence, the permission structure has to be modified. This can be achieved e.g. by the elimination of the link $(x2,x2)$ in R_{α_2}.

Then $a^3\{x5\} = \{x1,x2,x3,x5\}$.

5.2 Modal Logics

Detailed examination of the meaning of individual closures points at an interesting connection with modalities. For example a closure in the AIOU-modification of a topology describing the α-structures determines explicitly the set of subjects, i.e. it determines what is possible in certain situations. Similarly, different kinds of possibilities correspond to closures in other parts of the algebraic model (in permission and intention structures). It is obvious that, although formally the same in different parts of the model, the closures will express different grades of possibility according to the part of the model in which they appear. Apart from alethic modalities, there appear deontic modalities of permission and obligation. The intention structures are clearly connected with aims of subjects and this leads to yet another type of modality. However, each type of modality is not without relation to other types of modality and for this reason mixed modalities have to be introduced.

The algebraic method of McKinsey and Tarski, further extended by Lemmon /32/, /33/, provides a formal link between general topologies and modal logics. This approach can be extended to mixed modalities using results of /38/ on lattices of topologies and modifications of generalised topologies /39/, /15/.

6. Computer-Aided Design of Protection Structures

The simple example developed through the paper clearly demonstrates great complexity of the dynamics of protection structures. Our proposed mathematical models would be an academic exercise, devoid of the relevance to the real world protection and security problems, if they were not directly amenable to computer-aided design. As a matter of fact, our search for computer-aided methods for analysis and synthesis of protection structures, has (amongst other reasons) motivated our choice of mathematical techniques. We shall briefly outline some computational aspects of our mathematical techniques in the next lines.

6.1 A Metalanguage of Protection Structures and Theorem Provers

The ultimate aim is to design a machine theorem prover of statements about protection structure. Alethic modalities are sufficient for this purpose for we are not directly concerned with psychology of a designer. Semantic studies of logic suitable for expressing scientific, technological and legal problems, especially the recent development of a "Calculus of Problems" /40/ indicate that a S4 modal system may be sufficient for the study of the foundations of protection structures. Recently, very powerful mechanical proof techniques for modal logics have been developed, which are directly programmable on computers /10, p.12/.

6.2 Computation of Dynamics of Protection Structures

The algebraic model (cf. 4.3), which is formed by a hierarchy of sequential machines, presents usual computational problems of combinatorial character which are encountered in automata theory.

By forming closures on the protection automata, we select only the information which is pertinent to the given question, reducing enormously computational complexity. Dynamics of actions can be comprehensively researched using iterations and modifications /15/ of relevant topological spaces. Opting for these methods we eliminate exhaustive search for the sake of lattice structures and of iterations in lattices of topologies.

In the case where it is better to represent closures indirectly as possibilities in some modal logics, the techniques referred to in 6.1 above can be used. They are valid for very general modal systems /10/.

The mechanical proof techniques for modal and many-valued logics, which are of very recent origin, and therefore very little known and largely unused in computing, supplied the main motivation for our uses of powerful logics. Their importance can be highlighted by a quotation from Snyder /10, p.12/:

"Proving theorems within a given system of logic involves following a straightforward mechanical procedure. ... The high adventure of seeking clever strategies for deductive proofs, and the concomitant satisfaction of finding such proofs and being able to claim new theorems, are lost in the present set of formal systems. Instead, the adventure of doing logic ... lies in the development of a variety of systems of logic for a variety of tasks".

7. Conclusions

At the current state of the art the conclusions one may draw are still best summed up by Graham and Denning's final paragraph [17], "*Our preliminary work has indicated that the abstractions formed in the modelling process are useful in themselves, and that the model provides a framework in which to formulate precisely previously vague questions. We hope this discussion will motivate others to undertake additional research in this area. Much needs to be done*". The combination of interests represented at this conference seems peculiarly well suited to taking up this problem.

8. Acknowledgement

We are very grateful to Peter Facey of this Laboratory for many strenuous discussions on the topic of protection.

9. References

1. R.M. Fano, "The MAC system: a progress report", in M.A. Fass and W.D. Wilkinson (eds) Computer Augmentation of Human Reasoning, Washington: Spartan Books, pp. 131-150, 1965.
2. E.I. Organick, The Multics System, MIT Press, 1972.
3. E.E. David and R.M. Fano, "Some thoughts about the social implications of accessible computing", AFIPS FJCC, vol. 27, pp. 243-247, Washington: Spartan Books, 1965.
4. A. Westin, Privacy and Freedom, New York: Atheneum, 1968.
5. A. Miller, The Assault on Privacy, University of Michigan Press, 1971.
6. E.W. Djikstra, "Cooperating sequential processes", in F. Genuys (ed.) Programming Languages, London: Academic Press, 1968.
7. J.B. Dennis, "Modularity", in F.L. Bauer (ed.) Advanced Course in Software Engineering, Lecture Notes in Economics and Mathematical Systems, vol. 81, pp. 128-182, Berlin, Springer-Verlag, 1973.
8. O.J. Dahl, E.W. Djikstra and C.A.R. Hoare, Structured Programming, London: Academic Press, 1972.
9. G.E. Hughes and M.J. Creswell, An Introduction to Modal Logic, London: Methuen, 1968.
10. D.P. Synder, Modal Logic, New York: Van Nostrand, 1971.
11. B.R. Gaines, P.V. Facey, F.K. Williamson and J.A. Maine, "Design objectives for a descriptor-organised minicomputer", Proc. EUROCOMP 74, pp. 29-45, London: Online Ltd, May 1974.
12. F.K. Williamson, B.R. Gaines, J.A. Maine and P.V. Facey, "A high-level minicomputer", IFIP, Stockholm, August 1974.
13. B.R. Gaines, M. Haynes and D. Hill, "Integration of protection and procedures in a high-level minicomputer", IEE Conf., London, November 1974.
14. L. Kohout, "The Pinkava many-valued complete logic systems and their application to the design of many-valued switching circuits", Proc. Int. Symp. on Multiple-Valued Logic, IEEE, pp. 261-284, May 1974.
15. L. Kohout, "Generalized topologies: works of the Čech topological school and their relevance to general systems", Int. J. General Systems, vol. 2, Jan. 1975, to appear.
16. "The plan for information society", Final Report of the Computerization Committee of the Japan Computer Usage Development Institute, Tokyo, 1972.
17. G.S. Graham and P.J. Denning, "Protection - principles and practice", AFIPS SJCC, vol. 40, pp. 417-429, 1972.
18. G.J. Popek, "Protection structures", Computer, vol. 7, pp. 22-23, June 1974.
19. B.W. Lampson, "Dynamic protection structures", AFIPS FJCC, vol. 35, pp. 27-38, 1969.
20. R.M. Needham, "Protection systems and protection implementations", AFIPS FJCC, vol. 41, pp. 572-578, New Jersey: AFIPS Press 1972.
21. D.M. England, "Architectural features of system 250", INFOTECH State of Art Report on Operating Systems, 1972.
22. A.K. Jones, "Protection in programmed systems", Ph.D. thesis, Carnegie-Mellon University, 1973.

23. R.W. Conway, W.L. Maxwell and H.L. Morgan, "On the implementation of security measures in information systems", Comm. ACM, vol. 15, pp. 211-220, 1972.

24. H.L. Morgan, "An interrupt-based organization for management information systems", Comm. ACM, vol. 13, pp. 734-739, 1970.

25. M. Zelkowitz, "Interrupt driven programming", Comm. ACM, vol. 14, pp. 417-418, 1971.

26. E.A. Feustel, "On the advantages of tagged architecture", IEEE Trans. Comp., vol. C-22, pp. 644-656, 1973.

27. G.J. Sussman and D.W. McDermott, "From PLANNER to CONNIVER - a genetic approach", AFIPS FJCC, vol. 41, pp. 1171-1179, New Jersey: AFIPS Press, 1972.

28. R.M. Burstall, J.S. Collins and R.J. Popplestone, Programming in POP-2, Edinburgh University Press, 1971.

29. B. Wegbreit, "The treatment of data types in EL1", Comm. ACM, vol. 17, pp. 251-264, 1974.

30. B.R. Gaines, P.V. Facey and J. Sams, "An interactive, display-based system for gilt-edged security broking", Proc. EUROCOMP 74, pp. 155-169, London: Online Ltd, May 1974.

31. B.R. Gaines and P.V. Facey, "Some experience in interactive systems development and application", Proc. IEEE, June 1975, to appear.

32. E.J. Lemmon, "Algebraic semantics for modal logics I", J. Sym. Logic, vol. 31, pp. 46-65, June 1966.

33. E.J. Lemmon, "Algebraic semantics for modal logics II", J. Sym. Logic, vol. 31, pp. 191-218, June 1966.

34. S. MacLane, Categories for the working mathematician, New York: Springer, 1971.

35. J.A. Goguen, "Semantics of computation", in Proc. 1st Int. Symp. on Category Theory Applied to Computation and Control, Massachusetts, February 1974.

36. A.A. Arbib and E.G. Manes, "Foundations of system theory", Automatica, vol. 10, pp. 285-302, 1974.

37. A. Eilenberg, Automata, Languages and Machines, vol. A, New York: Academic Press, 1974.

38. E. Čech, Topological Spaces, Academia, Prague & J. Wiley, Interscience, New York, 1966.

39. K. Koutský and M. Sekanina, "Modifications of Topologies", In: General Topology and its Relation to Modern Analysis and Algebra 1, (Proceedings of the Symposium Prague 1961). Academic Press, New York & Academia, Prague, 1962.

40. P. Materna, "On problems (semantic study)", Rozpravy Československé Akademie Věd, vol. 80, sešit 8, pp. 1-62, 1970. (Published by Academia, Prague).