3. Guarantee Measurement and Composition

3 GUARANTEE MEASUREMENT AND COMPOSITION

Fear not to touch the best,
The truth shall be thy warrant.
Sir Walter Ralegh,
The Lie,
1595.

Since the measuring device has been constructed by the observer ...
we have to remember that what we observe is not nature in itself
but nature exposed to our method of questioning.
Werner Heisenberg,
Physics and Philosophy,
1958.

3.1. INTRODUCTION

Formal modelling techniques offer the benefits of precision of expression of the problem and its solution, as well as precision of elaboration of the chain of reasoning leading from suppositions to conclusions. However, there is a price to be paid for this precision, because of the necessity for casting the problem in a form amenable to analysis and compatible with available theoretical tools. This invariably seems to involve a certain amount of ''shoehorning'', occasionally taken to the extreme of discarding unmanageable ''toes'', with the result that the precisely expressed and precisely solved problem may not resemble the original problem as closely as we would like.

In contrast, it is possible to approach many problems in a less formal but more flexible way, relying on intuition and informal argument. The expression may not be as crisp, and the rationale not as objective, but by this method it is possible to deal with the practical aspects of a problem that the more formal methods cannot adequately comprehend.

We will develop such an approach to the measurement and composition of guarantee and assurance. We take as a fundamental principle of this work that the security issues that affect the trustworthiness of subsystems and of composite systems may be profitably divided into two groups, those relating to *guarantee* , which refers to properties of the system specification, and and those relating to *assurance* , which concerns our level of confidence that the

implemented system conforms to the guaranteed specification. We will begin by examining and refining these definitions to yield the most advantageous division of issues.

The methodology developed here will be influenced by the notions of guarantee contained in the published security criteria documents of several countries. This effort thereby serves the dual purposes of providing an alternate approach to the composition problems as well as providing guidance, in the form of a target based on official policy statements, for our formal modelling work.

3.2. GUARANTEE AND ASSURANCE

The preceding chapter examined the question of categorizing security concerns as being related either to guarantee or to assurance. In distinguishing the two categories we have tried to apply a consistent set of definitions. The statement of work issued for this research contained definitions of these terms, and required investigation of the concepts embodied in the definitions. A guarantee was defined as

> an assertion that a given security functionality is sufficient to satisfy the requirements of a specified security policy

while assurance was

> the level of confidence that a guarantee associated with a specified security policy is both valid and effective and that assurance is associated with both the quality and durability of the guarantee

In these definitions, the term *security policy* is considered to designate

> the set of laws, rules, and practices defining the protection of a system and the information it contains from unauthorized disclosure, modification, destruction, or removal and protection from the interruption of services that the system provides

Our objectives are to develop methods of measuring guarantee and assurance, and methods for deriving the guarantee and assurance of a composed system from the guarantees and assurances of its components.

3.2.1. Applicability of Terminology

In the security area it is conventional to make fairly narrow interpretations of such terms as ‘‘functionality’’ and ‘‘policy’’ that appear in the definitions above. Without detracting from the utility of these standard interpretations, we suggest that there is also value in taking a more liberal view of these concepts and the relationships between them.

The process of designing and building a system can be viewed as a succession of refinements, beginning with a statement of overall objectives and

ending with an instance of a mechanism constructed to achieve those objectives. The number of steps involved may vary, depending on the nature of the problem, the size of the project, or on the number of distinct groups involved in the process at different levels. It may even depend on who is doing the counting, as different observers may see varying numbers of informal refinement steps occurring in the same development process.

Still, it is possible to identify gross levels of refinement that should be acceptable to most readers. The process begins with a recognition of high-level **objectives** , usually as a collection of somewhat independent goals dealing with such issues as performance, cost, reliability, and control over the dissemination of information.

Consideration of these goals provokes and influences the establishment of **requirements** or **policies** , which are no more nor less than high-level design decisions expressing a large-scale strategy for achieving the specified objectives. The policy makers have some expectation, whether derived from

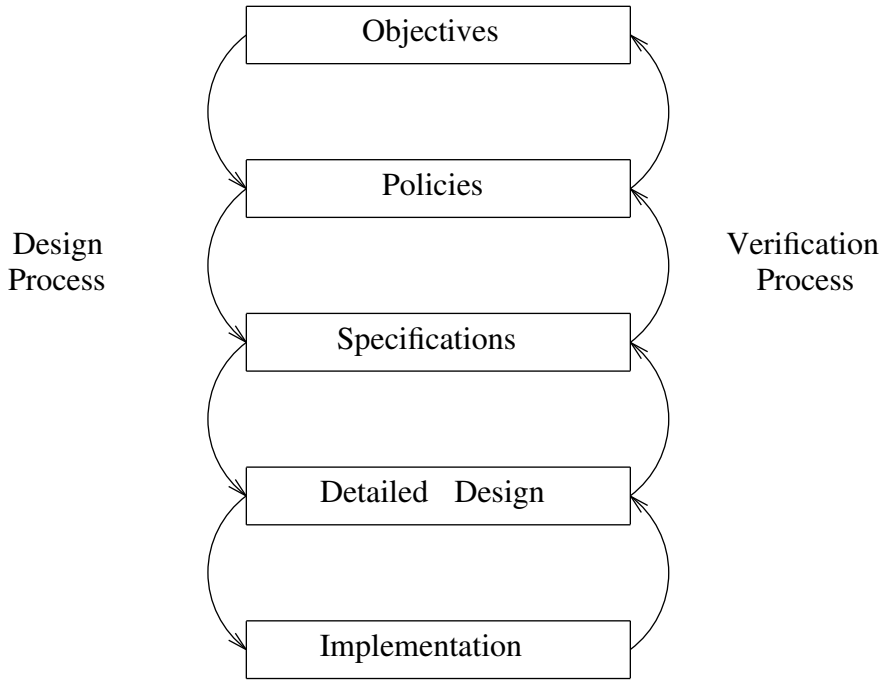

Figure 3.1: Refinements in System Design.

analysis or experience, that the enforcement of the policies will cause the goals to be met.

In the next step, often referred to as ‘‘high-level design’’ or, more increasingly, by such non-words as ‘‘architecting’’, a particular system is specified in sufficient detail to describe what it does, but not how it does it. This step is undertaken subject to the guidance provided by the policies and requirements established earlier. There is often a certain amount of compromise necessary here, since the several statements of policy and requirements may result in conflicting constraints on the specification.

The next refinement is the one conventionally known as ‘‘design’’, where an actual implementation is planned in detail. The specifications from the previous step define the target for this design, and success and failure of the design are largely defined in terms of meeting or failing to meet them.

Finally, the last step is the manufacturing, installation, operation, and upkeep of the system as a faithful rendering of the detailed design. Quality control, both over the raw materials and their processing, as well as the adequacy of the supplied documentation and of the prescribed maintenance procedures, are of prime importance here.

These five levels of specification are, in many cases, created by different organizations, or by different groups within an organization. They also deal with very different issues, and use different tools and notations in their production and presentation. Yet, the refinement steps and the relationships between them can be seen to share these qualities:

1) The refinements progress from the abstract to the concrete. Concepts at the higher level are more likely to be qualitative and subjective, while the lower level deals more with quantities and with specific objects.

2) Generalities at the higher level are made more specific at the lower. This may happen because a more narrowly defined range of problems is being addressed at the lower level which causes new application-specific requirements to be added, or because a more detailed view of the system makes visible low-level alternatives that were hidden at the higher level, or simply because the desired level of definiteness in the refined design requires that issues must be decided that were previously left open.

3) For each adjacent pair of levels there is some belief that the goal represented by the higher-level statement will be adequately met by its lower-level refinement. This belief may derive from experience, intuition, or confidence in the methodology employed, and may be augmented by the use of design verification techniques where such exist.

The cumulative effect of these refinements is to create a continuous derivation path from the statement of objectives to the concrete implementation, and to create a complementary ‘‘chain of belief’’ in the other direction. Although the language of the previously quoted definitions of guarantee and assurance seem to limit their application to the step between policy and specification, we believe it is both appropriate and useful to recognize that

the successful outcome of this process requires that adequate guarantees and assurances be established from the implementation level all the way up to the original statement of objectives.

One way to try to accomplish this goal is to extend the range of the verification methods. For example, in a software verification system one can represent both a software implementation of a module and a specification of that module. Theorem proving techniques are then used to prove that the module implements its specification. Such a system might be extended to also permit the description of a Bell-LaPadula style security model, which could make it possible to demonstrate that the specification conforms to a particular security policy.

Another method that has attracted some amount of attention in the security community in recent years is the development of what might be called security ''meta-models'', which model security policies and objectives and are used to demonstrate the relationships that hold between them. An early example of this is the model of Goguen and Meseguer [Goguen82] in which the objective is stated as a non-interference condition from high to low users, while the policy is represented as a set of conditions on states and transitions of a state machine. An unwinding theorem serves to connect the two and provide the desired assurance that enforcing the policy will achieve the stated objective.

Succeeding sections of this report will concentrate primarily on the versions of guarantee and assurance previously defined, those being the ones that relate specification to policy. However, the reader should keep in mind that the strength of the chain of belief depends on the quality of all of its links, and be aware that statements made concerning the relationship of specification to policy, and the guarantee and assurance that this involves, will also have validity in the more general setting of arbitrary pairs of levels.

3.2.2. Multiple Refinement Design

Acceptance of the Objectives—Policies—Specifications—Design—Implementation paradigm also sheds some light on a difficulty that is often encountered when attempting to distinguish between guarantee (or functionality) and assurance.

In some cases, the distinction is an easy one to make, and the separation of issues it supports is clearly beneficial. There is, for example, a clear difference in functionality between a system that offers mandatory access controls and one that does not. Also, the level of assurance provided by a group of implementors who are handed a specification and simply expected to follow it is clearly not equal to one in which a separate verification team formally or semi-formally establishes a correspondence between the specification and critical components of the implementation.

It is when trying to progress past these easily distinguished cases that we encounter aspects of system security that are hard to classify because they display characteristics of both guarantee and assurance, or where our definition of assurance fails to apply to factors that are commonly held to be assurance-related.

The conventional intuitive understanding of assurance is one that relates primarily to the methods used in the design and construction of a system, rather than the characteristics of the system so designed. This may not be the situation with our proposed definition of assurance, because it intimately ties an assurance to a corresponding guarantee.

That there should be some type of interaction is probably unavoidable, since providing an assurance is meaningless unless we also describe what is being assured. Still, we would like to develop a notion of assurance (and a measure of it) that operates independently of guarantee. It should allow us to make a meaningful comparison between the assurances of two different guarantees, based on the levels of effort, sophistication, and control exercised in the design and verification. In particular, if substantially the same techniques, procedures, and controls are employed in both cases, we should conclude that the assurance levels of the two guarantees are essentially identical, whatever the details of the policies and implementations.

Consider, for example, the problem of user identification and authentication. Passwords, ''smart card'' keys, and biometric devices represent a range of cost, ease of use, and effectiveness. A specification of the system will indicate an authentication mechanism and how it is to operate, and so the choice seems at first glance to be one of differing guarantees, and, in fact, probably different levels of a family of related guarantees.

Looked at from another viewpoint, however, the choice of which technique to use can be considered to be an assurance question, since it affects the reliability of an identification guarantee. This is a troubling observation, because one of the prime benefits that usually flows from separating a problem into its component parts is that understanding and analyzing the problem becomes much simpler. If we are unable to cleanly separate guarantee from assurance, these benefits will not be realized.

Figure 3.2 presents this situation diagrammatically, using the same five-layer breakdown of the design process as before. We assume in this analysis that the type of procedure acceptable for user authentication is considered a policy issue. There is a common element to all three approaches, which is represented in the diagram by a single system objective, that the system accurately identify users.

Let us assume that three systems are designed and built to the requirements of the three policies, and that ''substantially the same techniques, procedures, and controls are employed'' in the design and verification of the Policy—Specification, Specification—Detailed Design, and Detailed Design—Implementation correspondences for all three systems. Let us further

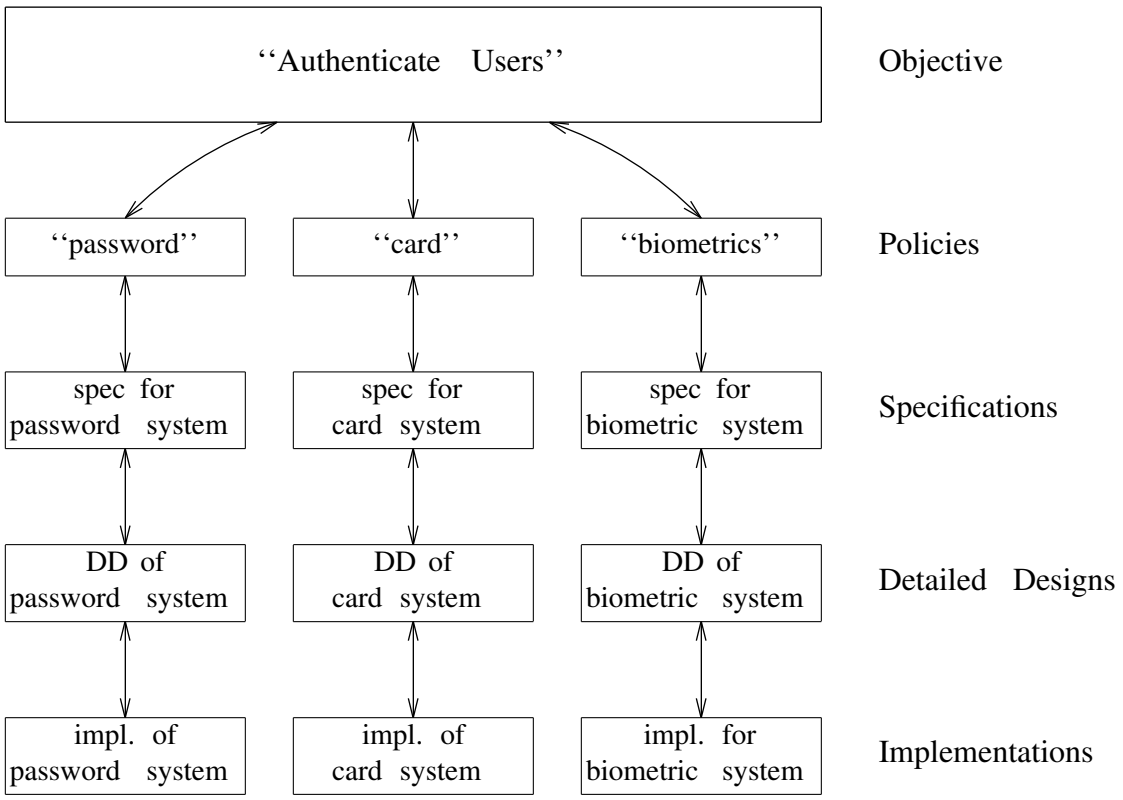

Figure 3.2: Differing Levels of Policy for a Single Objective.

assume that we have succeeded in developing a metric for assurance that behaves in an acceptable manner and which, in this instance, tells us that the assurances of all three Policy—Specification guarantees are equal, and similarly that the assurances of all three Specification—Detailed Design and Detailed Design—Implementation guarantees are equal.

Now, consider the overall guarantees of the three systems, that is, the composite guarantees from the Implementation to the Objective. Each is formed from a chain of guarantees leading to the same objective, and so are all of the form

Implementation *X* **satisfies the** *Authenticate Users* **Objective**

This common guarantee has different assurance levels across the three implementations, reflecting our assessment of the difference in reliability of the three authentication procedures. The source of these differences is the

Objective-to-Policy correspondence, which in each of the three cases consists of a guarantee of the form

Policy *X* **satisfies the** *Authenticate Users* **Objective**

but which, like the overall guarantee, has different assurance levels in the three cases.

This construction is consistent with the view that, in deriving three different policies each of which targets the same system objective, the framers of the policies must have been using different procedures in order to produce three results of different quality. In particular, they must have used an approximate definition of ''satisfies'', and varied the fineness of the approximation. Depending on the details of the definitions of guarantee and assurance, this variation may be evinced by differences in the measures developed for the assurances assigned to the guarantees, or by differences in both the guarantee and assurance measures.

At this point, we are inspired to make a short digression. Figure 3.2 describes an organization of hierarchical specifications that allows different levels of compliance with a common objective, in the same way that the published national evaluation criteria describe different evaluation levels. Also, figure 3.2 mirrors the criteria in one other respect, by having all of the ''looseness'', or variability, concentrated in the Objective—Policy correspondence. This seems to match the currently predominant method of accommodating multiple levels of strictness or compliance, in which looseness in assurances is not considered desirable in the lower layers, but is actively encouraged in the Objective—Policy pairing.

As mentioned earlier in this report, the current thrust of research into security modelling includes an effort to extend formal assurance methods used in lower layers to include all levels in the design hierarchy. It is likely that methods that are successful in dealing with the strictly defined satisfaction of the lower layer guarantees may prove to be inadequate at the Objective—Policy boundary, where more flexible forms of satisfaction seem to be warranted. New methods will be required, perhaps ''fuzzy''ones, that permit limited agreement of guarantee policies to objectives.

In fact, security modellers have already encountered a version of this problem. Sometimes, the higher-level policies may require such strict compliance with the objective that some looseness in the Policy—Specification pair may seem desirable. For example, a policy that requires the elimination of all covert channels is generally regarded as being unenforceable in any usable system. We will not further consider organizations that have flexibility in the lower levels. Degrees of adequacy should be decided at the policy level, and visible to the policy makers, because we believe that doing otherwise will lead to problems with observability of and effective control and regulation of this flexibility.

3.3. GUARANTEE MEASUREMENT

Although we find the concepts of guarantee and assurance useful as defined, a difficulty is encountered when we try to fulfill the prescribed requirement that guarantee must be measurable. A guarantee is an assertion of the form **A satisfies B**, where **A** is a functional specification and **B** is a security policy. Since an assertion is not a quantity, and therefore is not itself a measurable thing, we must instead measure some aspect of the guarantee.

Which aspect we choose to measure depends greatly on what the measure is to be used for. No guidance is available here from the statement of work, which requires that a measure exist but gives no hint as to what it should reflect or what its purpose might be.

3.3.1. Measurable Aspects of Guarantee

The most likely expectation is that the measure should somehow reflect the ''quality'' or robustness of the guarantee. This could be, for example, a measure of the amount by which the functional specification could be loosened while maintaining the validity of the assertion. However, anything relating to validity is, according to the definitions, a matter of assurance rather than of guarantee, so it is not possible to construct a guarantee measure along these lines without confusing the boundary between the two.

Probably, a more acceptable measure of the ''quality'' of a guarantee would be the simple measure of the ''quality'' of its security policy **B**. The security functionality would be ignored by such a measure, which would then mean that no ''credit'' would be received for functionality in excess of that needed to satisfy the policy asserted by a guarantee. This might be justified by arguing that the test of whether or not such excess functionality is significant should be whether or not it permits the satisfaction of a higher quality policy.

One possible rebuttal of this is that, if policy grades are defined as discrete steps, as in existing national criteria, then enhancements which approach but do not reach the next recognized step will still get no credit even though the enhancements do allow the satisfaction of a higher-quality policy. The answer to this objection involves recognizing that a significant improvement in achieved policy may require a ''critical mass'' of improvements in security functionality, or it may require simultaneous and synergistic improvements in more than one area. In other words, there is a discontinuous jump in quality when some particular functionality boundary is crossed. In these cases, it is entirely correct to grant no extra credit for a job partly done. Since we have no mechanical method of distinguishing this case from that in which quality increases continuously with functionality, it is prudent to treat all cases according to the more conservative rules.

A second objection concerns the situation where one specification clearly exceeds another, but in a way that is not quantified in the policy, or for which the policy specifies only minimum values. For example, a security policy may require that a computer system support at least four security levels and eight categories. It may be that these numbers are not modified in the requirements for any higher class of secure system. In some applications, an implementation that supplies more than the minimum may be more desirable than one that just meets the criterion. It might, therefore, be appropriate to attach some notation to the measurement that provides this information.

It is important, however, that any such auxiliary measurement must be considered orthogonal to, and subsidiary to, the main guarantee measure. Failure to do so would make it possible for a system that meets only a low-grade policy to, by exceeding the specified minima by higher and higher amounts, masquerade as an arbitrarily high-grade system, which is clearly undesirable.

3.3.2. Relationship to Criteria Documents

Having reduced guarantee measurement to a question of policy measurement, we must now determine under what conditions we can make supportable assignments of measures to different policies. This is a question that, in various disguised forms, can cause great confusion, although at least part of an answer can be easily provided here.

Various government-sponsored evaluation guidelines, including the Canadian TCPEC, the U.S. TCSEC, and others, specify sets of attributes which distinguish different levels of trustworthiness of information processing systems. In the U.S. document, trustworthiness is not separated into its guarantee and assurance aspects, but in more recently published criteria a separation is made. The separation tends to be between assurance and functionality rather than assurance and guarantee, so a modest amount of translation is necessary to relate the contents to our framework.

Let us consider the functionality classes of the (May 1990 Draft) Harmonised Criteria of France, Germany, the Netherlands, and the United Kingdom. There are ten defined classes in all, numbered F1 to F10. The first five of these are intended to correspond to the functionality components of the US TCSEC classes C1, C2, B1, B2, and B3/A1 [1], respectively.

In the terms employed in this report, F1 through F10 represent different policies, and assertions of compliance with them constitute different guarantees. The functionalities required by classes F1 through F5 are the most

[1] In the US document, the levels B3 and A1 are functionally identical. A1 is distinguished by the greater use of formal methods in the design and verification process, which is reflected in a higher assurance rather than a different guarantee.

instructive for us to study. They are hierarchical, because the demands made at each level are a superset of the demands made at the next lower level. At the same time, there is an obvious unity of purpose among them that is not found among the other five classes. It is clear that F1 through F5 correspond to different ‘‘quality’’ levels, in the loose way we have been attempting to use that word [2], and so should be granted different measures of guarantee by any metric we propose.

The classes F6 through F10, on the other hand, do not share any similarity of purpose. They are patterned after classes defined by the German Information Security Agency for use in particular applications or environments which emphasize different aspects of security, and so are not directly comparable with one another. It is unclear how the guarantee metrics assigned to these classes should relate to one another, or indeed, whether it is even possible for them to do so in any meaningful way.

The notion of increasing quality that we wish to capture is tied to the increases in functionality required as systems progress through the classes from F1 to F5. Each new added requirement may increase the scope or granularity of some control or audit function, or it may add specific constraints in an area that was more loosely specified at lower levels.

For example, consider the access control requirements. At level F1, the nature of access rights are not specified, and it is not necessary that they be applied to all objects in the system. There is a requirement that it be possible to completely deny users or groups of users access to objects, and that only authorized users can grant or revoke rights.

At F2, there is the added requirement that rights with respect to an object must be specifiable for individual users, and that some form of "read-only" access be available. Also, mechanisms (of an unspecified nature) must exist to control the propagation of rights.

With F3, the nature of the access controls becomes much more closely specified. They must include mandatory controls, of the familiar clearance/classification type, that apply to all subjects and all storage objects. These must be based on a dominance relation involving hierarchical levels and non-hierarchical categories, and must prohibit read-up and write-down.

In F4, there must be support for the grouping of access rights to define roles. At a minimum, the roles of ‘‘operator’’ and ‘‘administrator’’ must be definable. The final level, F5, adds only a few extra wrinkles: it requires separation between the roles of operator, system administrator, and system security officer, and that the access control system give the functionality of a full access control list implementation.

---

[2] Without defining it, to be precise.

These functionality classes have the property that any system satisfying one criterion will also satisfy all higher-numbered criteria. From this observation, it is reasonable to determine that the minimum quality guaranteed by the higher-numbered classes is no lower than that of the lower-numbered ones, but it is not possible to conclude that they are of strictly higher quality, simply because they are more restrictive. That judgement would imply that the extra restrictions actually progress toward some desirable goal, and such a statement can only be made after identifying and considering that goal.

The only creditable method of measuring the policies F1 through F5, then, must involve considering their relationships to their common goal. Those relationships are precisely the Policy—Objective level guarantees and assurances. For the sake of consistency, it is probably best that we consider these guarantees to be equivalent, since they do have the same objective, in which case the measures of the Specification—Policy guarantees are related to the assurances of the Policy—Objective guarantees.

To generalize the above, allow the guarantee whose quality is being measured to be between any two levels N and $N-1$ in the specification hierarchy, where level 1 is the uppermost or Objective level. Let us call this a *level N guarantee* . The quality of this guarantee is the quality of its level N specification, which is the ‘‘target’’ of the assertion in the guarantee. Although we could define this quality to be a function of the level $N-1$ assurance, the resulting definition would only be useful for comparing among level N guarantees that correspond to the same level $N-2$ specification.

This is pictured in figure 3.3 for two systems derived from the same ultimate objective. The level 4 guarantees, that is the guarantees between the Specification and Detailed Design, are indicated in the figure. Beside each guarantee is shown a dotted line representing its corresponding assurance. If we attempt to use the level 3 assurances (marked with asterisks) to gauge the quality of the level 4 guarantees, we will end up with incommensurate measures, because the two level 3 assurances do not refer to the same level 2 Policy goal.

Instead, the utility of the measure may be increased by relating it to the assurance corresponding to the *cumulative* guarantee from level N to level 1. In figure 3.3, these cumulative guarantees and assurances are indicated for level 3. The cumulative level 3 guarantees both refer to the same objective at level 1, and so the cumulative level 3 assurances that are associated with them are comparable measures of the quality of the alternative level 3 specifications. In turn, this means they are also valid measures of the level 4 guarantees.

In general, then, we choose to measure the quality of a level N guarantee by the measure of the cumulative level $N-1$ assurance. Determining that cumulative assurance need not necessarily require knowing the cumulative guarantees, because it may be that the cumulative assurance can be calculated directly as the composition of the level 2 and level 3 non-cumulative assurances.

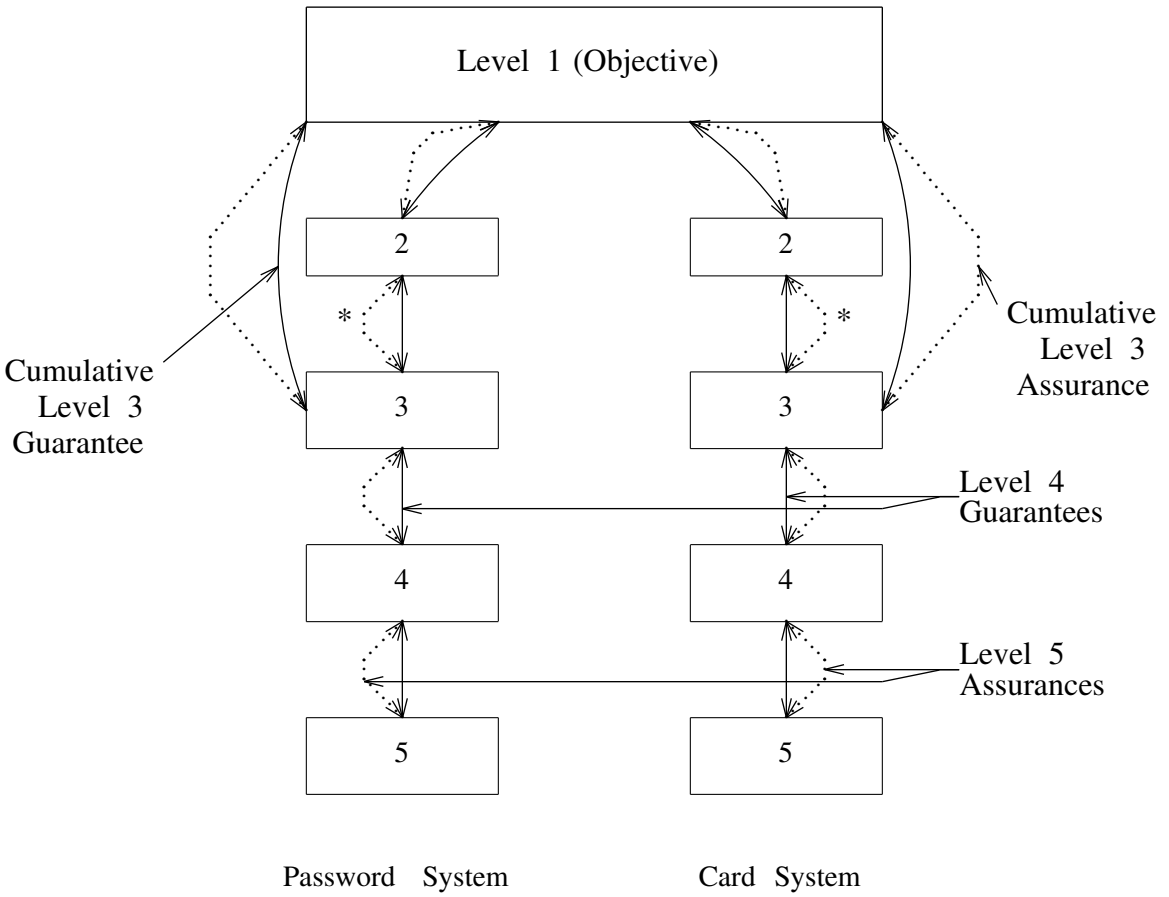

Figure 3.3: Guarantee and Cumulative Guarantee.

This alternative, however, does still require that assurances be determined for both levels 2 and 3, so that they may be composed to form the cumulative assurances. This may be difficult in cases where informal methods have been used at the higher design levels (notably, in the derivation of Policy from Objective). Similarly, when attempting to measure the quality of a level 3 or 4 guarantee, it may not be practical to accurately evaluate the assurance of an Objective—Policy mapping performed much earlier by a different group of individuals, and so it may only be possible to make a rough estimate of the assurance of that link in the chain. In that case, the accuracy of the derived guarantee measure may suffer, and so it may be

preferable to use a measure determined by the cumulative assurance from level N to level M, where $M > 1$.

To accommodate this situation, we will use the convention that the unqualified term ''guarantee measure'' will refer to a measure taken relative to the guarantee's level 1 objectives, but that, when desirable, measures may also be taken relative to a common lower-level specification, given that one exists. In that case, the measure should be called a ''relative'' measure, and the basis specification should be identified, if possible.

A precise statement of the form of a guarantee measure must wait until assurance and assurance measures have been considered, but in the interim we will simply state that the measure of a guarantee should be that component of the assurance ranking that relates to how strictly and how reliably the specified policies or mechanisms will enforce the desired behaviour.

3.4. COMPOSITION OF GUARANTEE

Setting aside for a time the final resolution of guarantee measurement, it is appropriate to consider the question of guarantee composition, which can be stated as follows:

> *Given some number of components that are composed to form a larger system, what guarantees for the aggregate can be derived from the guarantees of the individual parts?*

A guarantee consists of a combination of a specification and a policy statement, where the statement describes some characteristic or property of the specification. That being the case, there obviously exists for each particular composed system the straightforward approach of tossing out the policy statements, combining the specifications in accordance with the specific composition, and then analysing the resulting aggregated specification to derive whatever new properties can be discovered. To disqualify this approach, which is exactly the type of labour-intensive full system analysis that we are striving to avoid, the problem statement is amended with the addition of

> *... without significant manipulation of the specifications?*

It is certain that nothing significant can be said in the most general case, that is, for arbitrary components, compositions, and policies (or, equivalently, properties of interest). However, it may be possible to make nontrivial statements if one or more of these is restricted in some way.

One simple case where this is true is when the property of interest is the weight of a system. Weight can be determined knowing only the weights of the components, and is independent of composition. Power consumption of a composite electrical system can be similarly derived knowing only the corresponding properties of the components, but requires the slightly stronger stipulation that some rudimentary rules of composition be followed, such as connecting power only to power inputs, and not shorting or wiring

together outputs.

In the case of more complex properties, such as those relating to security, the answers do not come so easily. First it is necessary to understand the structure of systems, types of composition, and something of the nature of properties, before classes of each can be defined and investigated.

3.4.1. Abstract Systems

The study of *General Systems Theory* was inaugurated in the 1930s and '40s by Ludwig von Bertalanffy, a biologist, who was struck by the similarities in structure of problems arising in various fields and the corresponding similarities evident in their solutions. The now classical example is the analogy between a R-L-C electrical circuit and a mass-spring system with frictional losses, both of which, when disturbed, exhibit damped sinusoidal oscillations with exponentially decaying envelopes.

Von Bertalanffy suggested that the study of systems could be unified by abstracting away their identities, leaving only quantities and the interrelations that define their interactions. Consideration of these abstracted systems would lead to *the formulation and derivation of those general principles that are applicable to ''systems'' in general.* [3]

Major characteristics of the systems approach include recognizing the correspondences between similar systems in different areas that can be treated as a single abstract problem, and treating the abstracted system as an independently studied entity. Category theory, which has recently seen some use in formal security modelling, is a mathematical expression of these principles in which a system can be described by its variables or states, which form the *objects* of a category, and some relationships between them, the category's *morphisms* , and where the similarities between systems are expressed by mappings between categories, called *functors* . Formal manipulation of systems also invites contributions from graph theory, automata theory, and the theory of topologies.

Although the details of these formal models are not of interest here, we will take a brief look at the definitions and methods of classification employed in general systems theory, and consider the applicability of some of its results.

---

[3] [Bertalanffy68], pp. 32, 253

3.4.1.1. System Representation

When discussing the structure of a system, it is necessarily the structure of a model of the system that is being dealt with. And, inevitably, when the model is being conceived, some system aspects will be suppressed and others spotlighted. For any given system, the model chosen will depend on which set of characteristic quantities are under study.

Yet, independent of the system studied or of the system aspects described, a system can be considered as a collection of coordinated values of selected system attributes. In its most general form, this collection takes the form of a relation. The attributes may be simple values, in which case the model is a static one, or they may be time-dependent, in which case the model is dynamic. In addition, the attribute values, as well as the time values in the dynamic systems, may be considered either as discrete or continuous quantities. Of course, the same physical system may be represented by more than one kind of models for different purposes. For example, early in the design process a combinational logic device may be represented using a truth table or logic equations, yielding a discrete static model, while later on when signal propagation and transition times are considered it must be modelled as a continuous dynamic system.

In engineering use of systems there are usually some attributes which are externally controllable, and others which are only observable. This leads to the classification of attributes as *inputs* and *outputs* . The system description then takes the form of pairs of inputs and corresponding outputs.

It is also possible to introduce a generalized notion of