Architectures For Heterogeneous Reasoning On Interlinguæ

Dave Barker-Plummer and Mark Greaves Center for the Study of Language and Information Ventura Hall, Stanford University Stanford, California, 94305, USA

{dbp,greaves}@csli.stanford.edu

October 23, 1998

Abstract

The design of any computer system with a multimedia interface involves the designer in fundamental theoretical questions concerning the manipulation of information expressed in a variety of forms. Furthermore, if the system is to be intelligent then it must reason with and about the information that it represents. We use the term heterogeneous reasoning to refer to this task of reasoning with information presented in multiple forms. We define a heterogeneous reasoning system (HRS) as a composite deductive system which includes multiple component subsystems, each with its own syntax, semantics, and proof theory, and which also includes deductive rules which operate between the different subsystems. In the first half of this paper we provide a formal model which is able to distinguish several important types of heterogeneous reasoning.

Computer systems designers whose tasks involve the use of an HRS in an intelligent multimedia interface are typically faced with a choice of several implementation architectures. One basic architectural decision in the design of an HRS concerns whether to use an *interlingua*: a common underlying deductive system which is powerful enough to handle some or all of the inferential tasks of the HRS. While many multimedia systems are implemented using an interlingual architecture because of its conceptual simplicity, this is not the only architecture available. The impact of employing an interlingua in an HRS can be felt in various ways, from clouding the system semantics to increasing the complexity of deductions within the HRS. So, in the second half of this paper we discuss some of the implications and tradeoffs of adopting an interlingual architecture in an HRS.

Keywords heterogeneous reasoning, interlingua, logical theory.

1 Introduction

There has recently been an increase in interest in the problem of managing information that is represented in non-textual form. This interest has been spurred in part by the availability of sufficient computing power and network bandwidth to allow the construction of true multimedia applications. The methods with which a user interacts with and communicates information to a program have evolved far beyond the simple keyboarding and scientific plotting packages of thirty years ago. Now, computer programs can employ a variety of forms with which to present information to their users, including formatted text, charts, icons, graphics, animation, video clips, and sophisticated audio. Styles of input to programs have expanded from typed text to include mouse and pen movements, speech recognition, video feeds, and scanned and interactive graphics. In order to organize and manage interactions using these new forms of information, program designers have constructed complex multimedia interfaces, whose rules govern and give structure to the dialog between programs and users.

It is undoubtedly the case that these multimedia interfaces have increased the efficiency with which people use computer systems, and broadened the range of tasks for which computers are suitable. The ability to present and incorporate information in multiple forms has allowed the communication between programs and their users to be much more productive, as well as more natural. However, the proliferation of such interfaces has also raised some important and fundamental questions for interface designers. How can we quantify and be precise about the information that is presented by a diagram, photo, or segment of video? How can the familiar structures of formal logic, with which we are able to model the information contained in textual fragments, be generalized to encompass non-textual representations? And, with what techniques should a computer system draw conclusions with information that is given in non-textual forms?

We believe that a prerequisite for answering such questions, and for understanding the behavior of multimedia systems generally, is a theory of how non-textual representations can carry information, and of how computer systems can validly combine information that is represented in different ways to yield chains of reasoning. The answer to the first question is a matter of the semantics of individual non-textual representations, and is not something we will pursue here. In this paper, we will be concerned with the second question. In particular, we are interested in general questions of architecture: how should we organize systems which are required to draw conclusions from information which may be presented in a non-textual form?

To introduce this topic, let us look at a simple non-computer example of the sort of multirepresentational inference we have in mind. Consider the problem of finding a hotel in an unfamiliar city, given a textual description of the location and a map. The problem solver is required to combine information that is expressed sententially, e.g., "the hotel is on the west side of Union Square," with information expressed graphically by the map, in order to solve the problem of finding a route to the hotel. Clearly, both the sentential and graphical information inform the solution to the problem, and equally clearly, we have strong intuitive (as well as practical!) notions of what constitutes correct and incorrect inference using this sort of information.

Following Barwise and Etchemendy, [4], we call reasoning of the preceding kind heterogeneous. We intend this term to encompass reasoning problems whose solution essentially involves the use of information that is represented in multiple different forms, whether or not such forms include traditional sentential languages. We define a heterogeneous reasoning system (HRS) as a composite deductive system which includes multiple component subsystems, each with its own syntax, semantics, and proof theory, and which also includes deductive rules which operate between the different subsystems.

Turning to computers for a moment, we note that the traditional model of software systems consists of two components: an interface to serve as a simple channel of communication with the user, and a backend within which any domain-specific intelligence resides. The obvious generalization of this architecture to multimedia systems involves multiple interfaces tailored to different media, each communicating with the backend. But what do we mean by systems with intelligent multimedia interfaces? At least, such systems delegate to their interfaces some of the domain-specific knowledge and reasoning that would traditionally be associated with the backend. Therefore, typical systems with intelligent multimedia or multimodal interfaces will contain multiple sites of reasoning in addition to their backends. Each such site may require an individualized representation of information, which further entails an individualized logical theory at that site. Finally, these systems require a global logical theory, with inference rules and semantic relationships involving multiple sites, so that information residing at individual sites can be validly reexpressed at other sites which require it.

Heterogeneous reasoning systems form a useful abstract model which may be used to discuss and solve various problems which arise in the design of multimedia systems. Creating an intelligent multimedia interface will generally involve implementing a species of an HRS, because the system will need to perform heterogeneous reasoning between the different representations available in the system. Further, any abstract HRS may be realized in many different logically equivalent implementations, and the choice of specific implementation architecture may have profound consequences for the final system.

We have two aims in this paper: to formally define heterogeneous reasoning systems, and, to use our formal definition to discuss the space of possible architectures for an HRS, and describe the ways that different architectural choices can impact the various properties of the HRS, and hence the intelligent multimedia interfaces based on it. Several properties may be of concern when considering appropriate architectures for heterogeneous reasoning. From a purely logical perspective, the constraining features of an architecture revolve around its ability to make all and only the valid inferences that are licensed by the information in the component representations. A cognitive scientist might wish to place additional constraints on the architecture which reflect current understanding of the properties of the brain. We take a third perspective in this paper, namely the concern of implementing computer systems which support heterogeneous reasoning. In addition to questions of logic, therefore, we will also be concerned with questions of the computational efficiency of various architectures of HRS.

One HRS architecture which we will look at in particular depends on the use of an *interlingua*: a single deductive system which is powerful enough to handle all of the inferential tasks required by the HRS. Paradigmatically, the use of an interlingua in an HRS entails the following problem-solving methodology: first, translate all of the relevant information from the different HRS representation systems into the interlingua, then solve the problem in the interlingua, and finally translate the solution(s) back into the appropriate system. Such systems appear to offer certain advantages to the system designer, including formal simplicity and various sorts of modularity. We will examine these claims.

This paper divides into two parts. In the first part, represented by sections 2, 3 and 4, we define what we mean by heterogeneous systems, give some examples, and outline some possible specializations of the definition which represent interesting points in the spaces of HR systems. In the second part of the paper, we focus first, in section 5, on some general problems with interlingual systems, and then, in sections 6 and 7, on two possible properties which an interlingual HRS might lack.

2 Heterogeneous Reasoning Systems

Before proceeding to a detailed examination of the arguments about interlinguæ, it will be useful to provide some general background about our conception of an HRS. Our interest in heterogeneous reasoning systems is motivated by two observations:

1. Much of the everyday reasoning that we engage in involves information that is given to us in non-sentential form. Every time we use a map, read a

facial expression, or sense a temperature, we are using information whose expression is nonlinguistic. Further, although our computer programs might be required by their design to render the information into a formal language in order to process it, there is no reason to suspect that this is generally the case for arbitrary cognitive agents. In fact, reasoning with non-linguistic objects seems almost mundane, except that:

2. There are no general standards for what it means for a piece of reasoning to be formally valid, and yet rely on non-sentential information. The edifice of mathematical logic rests on the classical definition of satisfaction, which is closely tied to the syntax of first-order languages, and is therefore of little help in this more general situation.

In the space between these two issues lies the study of heterogeneous reasoning.

Our primary research, and the focus of this half of the paper, involves approaching heterogeneous reasoning from the perspective of mathematical logic, working on developing precise and rigorous models of reasoning with nonlinguistic informative entities. Our formal characterization of an HRS relies on the notion of a representation system. By this we mean a system of expressions which admits a consequence relation among those expressions.

Definition 1 (Representation System) A tuple S = (E, R) is a representation system if E is a collection of expressions, and R is a collection of inference rules $\frac{e_1, \dots, e_m}{e_{n+1}, \dots, e_m}$, where the e_i s are members of E.

In general E will be specified by a grammar, or some other generative device rather than a simple listing of expressions, and R may also be expressed as a collection of schemas abbreviating collections of inference rules.

Representation systems can have more or less powerful proof theories. For example, the grammar of first-order logic provides us with a collection of formulae E, and the familiar inference rules (schema) of natural deduction or resolution would serve as a suitable R. Other systems could be supported by special proof techniques appropriate to that system, for example map-reading inference rules. Still other systems might have more impoverished proof theories. The proof theories for individual nonlinguistic systems, such as graphical representation systems, are a topic which we do not have the space to present here, but the interested reader should refer to [1, 6, 7].

When many different representation systems are used collectively to express information about a problem and its reasoning domain, we have a heterogeneous

¹This work is done within the context of the *Hyperproof* Project at Stanford's Center for the Study of Language and Information.

reasoning system, or HRS. Formally, an HRS is a collection of representation systems together with additional inference rules which sanction the inference from expressions in one representation system to expressions in another. It is these additional heterogeneous inference rules which provide the links between the information represented in the different systems.

Definition 2 (Heterogeneous Reasoning System (HRS)) An HRS is a tuple (Σ, H) , with

- Σ is a collection of representation systems, $\sigma_1, \ldots, \sigma_n$. We will denote the set of expressions of σ_i by E_i and the inference rules of σ_i by R_i .
- H is a collection of heterogeneous inference rules $\frac{e_1,\ldots,e_n}{e_{n+1},\ldots,e_m}$. The only constraints on the e_i is that they each be drawn from the expressions of some σ_i , and that they are not all drawn from the same σ_i .

This notion of heterogeneous inference system is very general. We will give two examples of such systems in section 3, and discuss some important specializations of an HRS in section 4.

3 Examples of Heterogeneous Inference Systems

In this section we describe two example heterogeneous reasoning systems. The first, Hyperproof, is a completed, implemented computer system which allows users to formally reason about a blocks world by using both a diagram of the blocks in that world, and a set of first-order sentences which describe the world. The second example is drawn from the field of electrical engineering. It is not implemented as a computer program per se, but involves an identifiable heterogeneous reasoning system in common use.

3.1 The Logic of Hyperproof

Hyperproof [3] is a computer program which is part of a courseware package for teaching logic and analytical reasoning at the introductory undergraduate level. For the purposes of this paper, the pertinent feature of the program is that it provides an environment for completely formal reasoning which involves information expressed both textually and graphically. Hyperproof's domain is a fairly typical blocks world, consisting of blocks with various properties placed on a checkerboard. In support of reasoning over this domain, Hyperproof implements two interacting representation systems: a diagrammatic system which can

represent the world graphically and support certain kinds of graphical reasoning, and a linguistic system which supports first-order representation and reasoning concerning this same world. Hyperproof is therefore an implementation of an explicit HRS which contains exactly two representations.

The design of the linguistic system is quite traditional; it includes a first-order language with equality similar to that of Tarski's World [2], plus a complete system of inference rules and a matrix-method theorem prover to allow the familiar kinds of reasoning between these sentences.

The graphical component is more interesting. The blocks world can be populated by up to twelve objects. Each object is located somewhere on the checkerboard, is one of three shapes, is one of three sizes, is assigned one or more names, is either happy or not, and either likes or dislikes each of the objects on the board. The diagrammatic language allows the user to specify any subset of the above properties for each of the objects, and so allows the depiction of partial blocks worlds. For example, if a particular block is drawn beside the checkerboard, the diagram is interpreted as not specifying a location for that block, although that block is actually positioned somewhere on the checkerboard. Similar conventions allow the diagram to be indeterminate about a block's size, shape, happiness, names, and state of mind towards other blocks.² Also, the diagrammatic system enforces a simple proof theory concerning permissible modifications to the diagram.

Hyperproof's sentential and diagrammatic representation systems were specifically designed not to be coextensive. That is, there are certain facts about the blocks world which can only be represented by using the diagrammatic representation, and other facts whose only representation would be sentential. For example, the sentential language has no ability to express absolute block positions on the checkerboard. One can use the sentential system to express that a is to the left of b (as LeftOf(a,b)), but not that a is on a particular square, because the language lacks the appropriate predicates and constants. In order to express precise position facts, the user is required to employ the diagram, where a block can be depicted as occupying a particular square. Similarly, as is often the case with diagrammatic representations, Hyperproof's diagrams cannot represent general disjunctive or negated information, and so complex logical relationships have to be expressed linguistically. These complimentary sorts of expressiveness in Hyperproof's two representation systems make it possible (and indeed, quite easy) to pose problems whose solution essentially involves heterogeneous reasoning. An example proof in the Hyperproof system is shown in figure 1.

²It is a convention of Hyperproof's diagrammatic representation system that all of the blocks in the world will be represented by some token; that is, the diagram cannot be indeterminate about the number of blocks in the world. This restriction functions simply to make the proof system easier for students to use, and is not required by Hyperproof's underlying logical theory.

Figure 1: An Example Hyperproof

The HRS which is implemented in Hyperproof includes heterogeneous inference rules spanning the two representation systems. These rules allow inference of new sentences from diagrammatically expressed information, and new diagrammatic information from sentences. The **Observe** inference rule allows the inference of any sentence whose content is represented by the diagram, and therefore if the current diagram depicts only tetrahedra (and no object of unknown shape), then the sentence $\forall x Tet(x)$ can be deduced. There is also a companion rule to **Observe**, called **Apply**, in which the diagram may be modified on the basis of information expressed sententially. Finally, there are various rules which require premises drawn from both representational systems.

Hyperproof is an interesting example of an HRS because not only is the user's experience of the logical system essentially heterogeneous, but there is no common internal representation of the information present in the problem.³ Many users of the system assume that inference within the system occurs by translating the diagrammatic and sentential information into some central representation (often assumed to be an extension of the sentential system), and that all inference occurs within this common system, with the results translated back to the appropriate user- accessible representation. This architecture, for which we later introduce the term paradigmatically interlingual, would certainly be a familiar and feasible implementation strategy for Hyperproof. It would, however, conflict with one of the theoretical goals of the Hyperproof project to try and make the internal system logical design match the external logical systems it incorporates, and so gain real experience with thoroughply heterogeneous reasoning. Further, it is a commonplace that different sorts of representations facilitate different sorts of inference, and that using inappropriate representations can dramatically complicate the reasoning that would otherwise be quite simple. Hyperproof was designed to be a vehicle for investigating these computational issues. Adopting an interlingua would commit Hyperproof to homogeneous proofs internally, and would have made it difficult to use Hyperproof as a tool for investigating these issues.

³We are ignoring a potentially confusing issue here, namely the level of abstraction at which we view the system. Since Hyperproof is a computer program, all of the information that it manipulates is represented in the form of binary encoded data (or electric currents if you view the system at a sufficiently low level of abstraction). We take the view that the correct level of abstraction is one in which the computer programmer builds data structures which represent first-order sentences, on the one hand, and diagrams on the other. At this level of abstraction, the only way in which the representations can interact is by the explicit use of a heterogeneous inference rule.

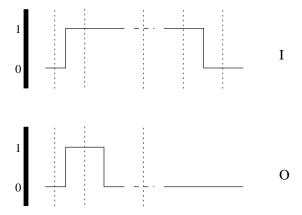


Figure 2: Timing Diagram of a Unit Pulser

3.2 Reasoning about Circuit Design

Electrical engineers have developed a number of different types of representations, each of which is designed to display information relevant to a different aspect of the circuit design task. The representations with which we are most familiar include circuit diagrams, which describe the logical components and interconnections in the circuit; timing diagrams, which describe the time properties of signals on wires in the circuit; and state machines, which describe the abstract logical behavior of the desired circuit. Experienced engineers employ reasoning involving all three of these representations in the course of designing a circuit.

Barwise and Etchemendy give the following example of these systems in use in [4]. Figures 2–4 represent three views of a unit pulser, a simple circuit whose output is usually low, but which goes high for one unit time interval each time there is a low-high transition on the input. The temporal aspect of this circuit's behavior is typically represented via the timing diagram of figure 2.

Another way of describing the behavior of this circuit is as a state machine. At this level of description (figure 3), the device has two states. The device stays at the rightmost state in the figure while its input is low, and transitions to the leftmost state while outputting a one ("pulses") on high input. This leftmost state is stable with no output until the input goes low, in which case the machine returns to the rightmost state.

The final description of the device that we will consider describes the logical components of the circuit. Here we see that the circuit involves a delay unit, which copies its input to its output after a delay, connected to a not-gate and an and-gate.

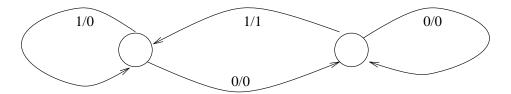


Figure 3: State Diagram of a Unit Pulser

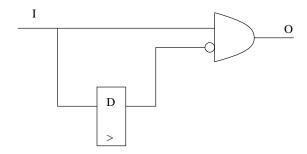


Figure 4: Circuit Diagram of a Unit Pulser

Each of these individual representations of the behavior of the circuit is designed to allow the user to reason about specific features of the circuit, and was developed for that reason. However, in order to draw conclusions about the circuit as a whole, it is often necessary to combine information that is represented by each of the diagrammatic systems. For example, the circuit diagram does not give precise timing information, and neither the timing diagram nor the circuit diagram can give gate-based information. Reasoning involving the possible failure modes of the circuit requires information from all of the representations. Thus, we can view this collection of individual representations, augmented with additional inference rules for transferring information between representations, as an HRS.⁴

4 Mapping the HRS Space

In this section we formally describe two specializations of the notion of HRS. In the first of these heterogeneous inference is restricted to only moving information between pairs of representation systems, and so we will call this a *conversion HRS*. In the limiting case, inference rules in a conversion HRS simply allow for the *translation* of information, while inference proper is performed within

⁴Kathi Fisler has worked extensively on formalizing the kinds of valid inferences that are possible with this heterogeneous reasoning system. Her work is reported in [5].

the individual representation systems.⁵ The other class of HRS we identify is the *interlingual system*. In these systems, the only heterogeneous inference rules involve a distinguished system, the *interlingua*, via which all heterogeneous inference is performed.

This discussion can be thought of in one of two ways. On the one hand the identification of these specialized, well-behaved, classes of HRS will enable the explorer to classify and therefore understand new heterogeneous reasoning systems when they are encountered. From our point of view, a more illuminating perspective is that these specializations represent possible design techniques. If the task at hand is to formalize a previously informal HRS, it might be possible to map the informal system into one or more of these more specialized classes of system. The choice of formalization should have no impact on the theorems of the system, of course, but choosing to design a system interlingually (for example) rather than as a conversion or other type of HRS may have computational or other consequences. It is these consequences that we investigate in sections 6 and 7

4.1 Conversion Heterogeneous Reasoning Systems

Given our characterization of an HRS, we can define a heterogeneous deduction as a deduction sequence in the HRS which contains the use of at least one heterogeneous inference rule. Any heterogeneous deduction in a given HRS can be divided into subchains of two types: deduction chains which occur entirely within the component representation systems, and applications of heterogeneous rules.

The generality of our definition of an HRS allows heterogeneous inference rules to have as their assumptions expressions from any number of representation systems, and to allow the deduction of a collection of expressions again drawn out of any number of different systems. We believe that it will frequently be the case that this generality will not be used, and thus that the corresponding restriction of the general notion of HRS is noteworthy.

Definition 3 (Conversion HRS (CHRS)) We call a heterogeneous inference rule a conversion when it involves expressions from just two representation systems, with the conclusion expressions all belonging to one, and the assumption expressions all members of the other.

A heterogeneous reasoning system (Σ, H) is a conversion system just in case all of the members of H are conversions.

⁵There is, of course, an interesting philosophical question here about whether translation is a species of inference.

In a conversion HRS, heterogeneous inference is used to transfer information from one representation system to another. We note that a conversion HRS allows the adoption of an inference technique which we call fractioning. Fractioning occurs when all system deductions can be partitioned into chains of homogeneous inferences (using members of R_i), and chains of heterogeneous inferences using the conversion heterogeneous inference rules to translate information from one representation system into another. Fractioning functions as an organizing principle over the search space of possible inferences, and often allows for a cleaner system logical design.

As an example of a conversion rule, consider Hyperproof's **Observe** rule described above. **Observe** is the simplest kind of conversion, involving one assumption expression, the diagram in force at that point in the proof, and one conclusion expression, the (single) observed sentence. The **Apply** rule, however, is not an example of a conversion rule, since it applies a collection of sentences to a given diagram, to produce a new diagram. That is, the assumption expressions for **Apply** are themselves heterogeneous between the sentential and diagrammatic representation.

4.2 Interlingual Heterogeneous Reasoning Systems

As we observed above, one familiar approach to the management of information expressed in many representations is to translate all of the information into a common representation, and then to perform inference in this common representation. We call this approach to designing an implemented HRS the paradigmatically interlingual approach, and the common representation an interlingua. This design strategy has a certain easy appeal, and is the core architecture of many of the heterogeneous database applications available today. In addition to being the "obvious" generalization of the basic interface/backend methodology, the paradigmatically interlingual approach also appears at first to defuse several logical problems that might arise from the management and manipulation of heterogeneous information. In particular, using an interlingua seems to provide a focal point for issues relating to the system's data semantics, computational properties, and correctness and consistency of its deductive rules. The second part of this paper will be devoted to examining the details of interlingual systems with respect to some of these claims, and contrasting them with their presumed benefits. Let us begin by defining more precisely an interlingual HRS and its stronger cousin, the paradigmatically interlingual HRS.

We formally define an *Interlingual HRS* in the following way:

Definition 4 Interlingual HRS (IHRS) An HRS (Σ, H) is interlingual if it is conversion system, and if further there is a distinguished representation system $I \in \Sigma$ such that all of the inference rules in H either draw all of their

assumptions from I or make all of the their conclusions in I.

Interlingual systems may distribute the computational burden of inference in different ways. For example, it may be possible to make the same deduction wholly within some surface representational system, or wholly within the interlingua. One extreme case occurs when the interlingua is used only as medium through which information can be translated between the other representational systems. In this case R_I will empty. We will call these trivially interlingual systems. At the other extreme, the HRS may involve a number of surface systems which do not themselves support inference, and so the only way to use the information expressed in these systems is to translate it into the interlingua. We call these systems paradigmatically interlingual.

Definition 5 (Paradigmatically Interlingual HRS (PIHRS)) An HRS (Σ, H) is paradigmatically interlingual if it has an interlingua σ_i , and R_j is empty for all $j \neq i$.

Trivially interlingual and paradigmatically interlingual HRS's are the endpoints on a spectrum of architectures which divide the burden of inference among the various deductive systems operative in an HRS. We will refer to any HRS which places a significant portion of inferential burden on a single, often non-surface representation as an essentially interlingual HRS.

The Internet's World Wide Web provides a simple example of how a multimedia interface can be structured as a PIHRS. One of the Web pages accessible from the authors' home site includes a street map of the town of Palo Alto, with several local restaurants marked. Clicking on any of the restaurant icons causes the web browser to translate the click into an HTML imagemap query and forward the query to its specified server. The server uses the query to deduce the restaurant intended by the click, and sends back to the browser an HTML description of that restaurant's offerings. Finally, the browser translates the HTML stream from the server into the formatted text which the user sees on her screen. Now, analyzing this system as a PIHRS, we can identify two surface representations with empty proof theories – the (graphical) location map and the (textual) description. The language of I, the interlingua, is HTML, and the proof theory of I is implemented by the server. Finally, the rules of H which link I to the surface systems are provided by the semantics of the browser.

We emphasize that there is nothing special about a representation system per se which makes it an interlingua. Such systems are just representation systems like any other; it is the relationship that these systems bear to the other representational systems in the HRS which identify them as interlingua. They may even be surface representational systems. However, the usual intention behind a paradigmatically interlingual HRS is that the interlingua serves "behind

the scenes" as the common currency of exchange between the different surface representation systems. Often, the interlingua will not be especially suitable for human use. This can happen, for example, when the interlingua is a first-order or database language specifically engineered to facilitate the automatic computation of certain classes of desirable inferences.

5 Questioning Interlingua

Having set up a framework for an analysis of heterogeneous reasoning, we now turn our attention to the question of the costs and benefits of adopting the interlingual approach to heterogeneous reasoning. In [4], Barwise and Etchemendy remark that the use of interlingua in heterogeneous representation systems is theoretically unnecessary and in addition, that the practical utility of interlinguæ is debatable. The first of these claims is a matter of logic, and we agree with it. Indeed, the Hyperproof program, previously described in section 3.1, is an implemented computer-based HRS which contains no interlingua. This example is sufficient to refute any general claim about the indispensability of interlinguæ in an HRS. However, their second claim deserves a fuller exposition. In their discussion of interlinguæ, in the context of the digital design example referred to above in section 3.2, Barwise and Etchemendy state:

Imagine trying to design a hardware system in which all three forms of information were represented in a single system. Experience shows that such a system would lose the clarity, crispness and utility of the systems that have developed in practice. To combine these into one system, the mappings from syntax to semantics would have to be complicated to an extent that they would lose much of their homomorphic character. The more complicated this mapping becomes, the closer we come to a notational variant of a linguistic scheme of representation, thereby losing the very representational advantages inherent in a homomorphic system.

For Barwise and Etchemendy, a representation is homomorphic if the representation is structurally similar to the domain that it represents, a point to which we will return in a later section. We share the intuitions behind the quoted observation about the possibility of common representations of the information presented in the three systems to which Barwise and Etchemendy refer. However, let us consider a contrasting example in order to evaluate the basis for these intuitions, and the generalizability of the conclusions based on them. Here a simple heterogeneous reasoning task for which the right interlingua seems clearly beneficial. Consider the problem of determining the best

route to drive from the Stanford Linear Accelerator Center (SLAC) to Stanford's Main Quad. Imagine that the required information for this task is given by the following representations: a labeled connectivity graph of the streets on Stanford Campus, but with no text on the map identifying places by name; and a list of sentences stating the coordinate position of different places on campus, e.g., "SLAC is at (130,90)."

Figures 5 and 6 show this representations of the problem:

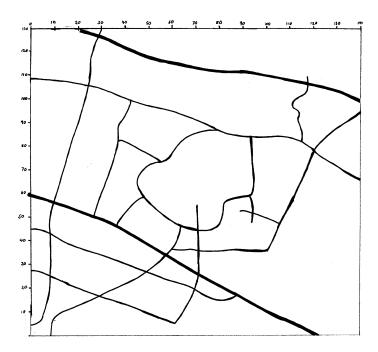


Figure 5: The Streets of Stanford Campus

Buck House	(120,85)
SLAC	(130,90)
Hoover Tower	(55,60)
Main Quad	(70,60)
Stanford Stadium	(50,45)
Medical Center	(85.55)

Figure 6: Location Information

By knowing constraints of legality (speed limits, say) imposed by the type of streets, we can reason to an acceptable route (perhaps one which meets a time constraint) from SLAC to the Main Quad. In this task, each representation system supplies information crucial to the inference. Yet, it is also clear that, given the various perceptive and cognitive capabilities that humans possess, our deductive task would have been made far easier if instead we had adopted a graphical interlingua in which the place locations were recorded on the map along with the streets, as in figure 7. Contrary to the prediction made above, this interlingua does not sacrifice the degree of homomorphicity of the individual surface representations. Neither do we end up with an interlingua that is more language-like than both of the surface representations.

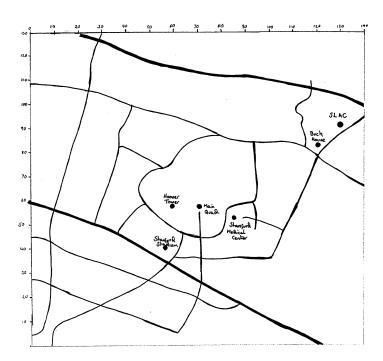


Figure 7: An Effective Interlingual Representation

What lesson shall we draw from these two examples? The route-finding example appears to confound any absolute claim that interlinguæ in an HRS will necessarily lack homomorphic qualities relative to the individual systems that they subsume, and demonstrates that the disadvantages of the use of interlingua in an HRS are not as straightforward as they might at first appear. Nevertheless, Barwise and Etchemendy's quote gives voice to a powerful intuition about the

form and role interlinguæ play within a typical HRS. In the following sections, we consider some issues which might be important in determining whether a particular HRS application might or might not benefit from the adoption of an interlingual architecture. Our discussion is organized around the issues of "clarity, crispness, and utility" which are referred to in the quote above; while we are confident that Barwise and Etchemendy did not intend these particular words to exactly embody their considered views, they nevertheless provide a handy starting point for the sections that follow.

6 Semantics

One possible problem with the adoption of an interlingual architecture for a heterogeneous reasoning system is that the semantics of expressions in the interlingua can become very complex to compute. That is, the complexity of the task of performing the semantic mappings which hold between the representation and the domain represented may be increased by adopting an interlingual representation capable of representing all of the information representable in the surface systems. To the extent to which an agent has difficulty with that operation, we will take the complexity of the representational system to be problematic.

This agent-relative character points up the notion that for representations used in this way, whether or not a particular semantic mapping is "complex" is primarily an observer- and agent-based notion, and is only derivatively related to traditional mathematical measures of the complexity of a signal or of a computation. Consider two representations of the same musical performance – the conductor's annotated score, and a printout of the binary encoding used to create a CD of that performance. For someone trained in music, the score is far more clear about the nature of the music than the digital dump. For someone who cannot read music, both representations are opaque, though the rendition of the CD data through an appropriate encoding device (a stereo and speakers) would be definitive.

It is therefore difficult to say that a representation is clear or unclear absolutely, but rather the agent that is to perceive the representation must be taken into account. In particular, since we are concerned about the question of interlinguæ in an HRS, we cannot assume that the relevant measure is whether human users find the meaning of expressions in the interlingual representation difficult to compute. Going back to the circuit design example, if we intend

⁶It is possible that this is what Barwise and Etchemendy mean when the say, in the quotation cited above, that an interlingua for the circuit design problem would lack *clarity*. Whether or not this is their intention, we consider the issue of the complexity of the semantics of the representation(s) to be central to the adoption of an interlingual architecture.

our hypothetical interlingua to be targetted at a computer-based verification system, then the claim that an interlingua will be much less clear (for such an agent) is not obvious. And, while the example is plausible from the point of view of human agents, a unified representation of our three types of information about circuit design is still conceivable, despite our not having developed it yet.

Lets assume that the claim is that the semantics of an interlingual system is less clear for humans than a corresponding non-interlingual system. This claim is plausible if what is being compared is the semantics of the interlingual system with the semantics of each surface system separately. But this is really like comparing apples with toothpicks. To make a fair comparison, we need to compare the semantics of the interlingual system with the "semantical sum" of the constituent systems in the non-interlingual case. This is much more difficult, since we have no clear definition about the "semantical sum". The most generous reading of this notion is something like the maximum complexity of the constituent systems, but we believe that the clarity claim fails even for such a notion.

We again appeal to the problem of circuit design. We can imagine a heterogeneous system in which each individual surface system is designed specifically to facilitate reasoning about a different type of high-level component: floating point dividers, timers, instruction pipelines, and so on. We perceive the semantics of such objects to be very complex. At base however, all of these components are built out of a single type of object: the transistor, which has a very simple semantics. If we chose the language of transistors as our interlingua, then the semantics of the resulting system would certainly be very simple relative to that of any of the surface systems, despite the small informational grain of the interlingua.

And, reasoning about the logical behavior of a complex microprocessor by using a transistor-level representation instead of a higher-level representation would not necessarily be simpler, even though the semantics of the former representation is clearer in our sense than that of the latter. This is because the description of such a device at the transistor level must be much more detailed than at the higher level. As a psychological matter, we might still prefer to be able to reason with a representation with complex semantics than with a simple semantics because of the benefits of modularity and abstraction that this representation allows. A computer, on the other hand, is often more easily designed to work with representations with simple, uniform semantics.

The conclusion of this discussion, therefore, is that while the semantics of an interlingua may be more complex than the individual representation systems between which it mediates, this is not necessarily the case. Further, the increase in complexity may or may not be a good thing, depending on the reasoning agent and context.

7 Proof Theory

Another important feature of heterogeneous reasoning system is the complexity of performing inference within the system. Different representations of the same domain will make different features of the domain accessible, and the general question of whether a particular representation system is useful is therefore linked to the identification of a class of inferences that are likely to be performed, as well as the computational properties of the representation per se. Simply put, a useful representation has to match well with the reasoning context and the problems to be solved. So, parallel to the question of whether the semantics of the representation is clear, the ability to perform inference using a representation is dependent on the the abilities of observer. Whether a particular representation makes inference easier depends on who is doing the inference. Different perceptual abilities on the part of observers may make a single representation good for one and bad for another. See, for example, the work reported in [8], on individual differences in the use of Hyperproof.

It seems to us that this issue is the one which designers of intelligent multimedia systems should pay the most attention. The efficiency with which inference may be carried out by the computer will at least in part determine the usability of the resulting program. The choice of architecture within which to perform reasoning tasks is most visible to the user at this point.

7.1 Hyperproof and Homomorphic Representations

The Hyperproof blocks world diagrams make it convenient for humans to reason about, for example, the Adjoins relation. In Hyperproof, a block x adjoins another block y if x is located on a square which shares a side with the square upon which y stands. Adjoins is a symmetric relation, as is immediately obvious to a human observer employing Hyperproof's diagrammatic representation. Not only is this property of the Adjoins relation made obvious, but it is impossible to diagrammatically represent a situation in which block a adjoins block b, but block b does not adjoin block a. This feature of the representation not only facilitates the inference that Adjoins is symmetric, but almost forces it. On the other hand, the sentential representational system of Hyperproof's HRS does not have this feature. One can assume Adjoins(a,b) as a hypothesis of the proof, but in order to infer Adjoins(b,a) one would additionally need a (linguistically-stated) axiom asserting the symmetry of the relation.

In a paradigmatically interlingual implementation of Hyperproof, we might adopt an interlingua which is based on an extension of the existing sentential system. But doing this would require that inference concerning the *Adjoins* relation which the graphical representation computes "automatically" would

be made explicit in the interlingua. This would appear to argue against the adoption of a sentential interlingua for Hyperproof on the grounds of what we are calling proof theoretic efficiency.

7.2 Granularity

In the light of the Hyperproof example above, the claim that an interlingual system necessarily decreases efficiency looks appealing. We believe that this appeal rests on two assumptions:

- 1. That the "grain size" of the information represented in the interlingua is no bigger than the "grain size" of the information represented in the surface systems. This is likely because the interlingua must be capable of representing any information which is representable in any surface systems.
- 2. That the "grain size" of inference within the interlingua will also be the smallest of the grain sizes of inference within the surface representation systems⁷. This is likely because the interlingua must be capable of representing any inference which is representable in any surface system.

If these assumptions were correct, an argument against the utility of the interlingual system would rest, first, on the observation that the space of possible inferences in the interlingua must be larger than that of the space of inferences in each of the surface representations, and second, that this increase in search space would lead to a more complex and time-consuming search for appropriate chains of inference when reasoning with the system.

We find the first of these assumptions unremarkable, but the second problematic. The intuition behind this assumption is that the interlingua must be able to validate all of the inferences possible in and between each of the surface systems, and this is true. But, there is nothing to prevent the interlingua from making use of inference rules which also make large inferential jumps. In fact, we see this as likely, since the interlingua represents "all in one place" information previously distributed among different surface representation systems.

As an example of how this might work, consider a heterogeneous reasoning system in which there are a family of surface representation systems, each of which is capable of representing exactly two propositions: system n being able to represent propositions P_n and P_{n+1} . Imagine further that $P_n \vdash_n P_{n+1}$ in each system n. To infer the theorem of the heterogeneous system $P_0 \vdash_H P_{i+1}$,

⁷We do not have a precise formalization of the notion of "grain size", either of information or of inference, relying on the readers intuition.

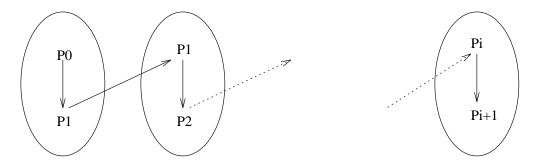


Figure 8: Non-Interlingual Approach

it would be necessary to start in system 0, with P_0 make a single inference in that system, translate the result into the next system, and iterate this process until system i is reached (see figure 8). Each inference step involves one inference and a translation of the new information into another system.

For such a heterogeneous system (which might be considered pathological, but which is certainly plausible) an interlingual system could be arranged to allow more efficient inference. Since the interlingual representation may represent all of the propositions P_k , we can deduce $P_0 \vdash P_i$ through exactly the same chain of inferences without incurring the overhead of translating between them (although of course we have to translate P_0 into the interlingua, and P_i from it). The heterogeneous system will be more efficient, ceteris paribus, since it avoids the plethora of translation steps necessary in the non-interlingual system (see figure 9).

An additional advantage is potentially available in the interlingual case, and that is that inference rules may exist which allow the inference of P_{i+1} from P_0 without going through the intermediate steps. This is possible because P_0 and P_{i+1} are both expressible in the interlingua, while in the non-interlingual case, the proof must proceed via the intermediate steps, because none of the intervening representation systems are capable of representing information from other non-neighboring systems.

7.3 Modularity

Another possible argument in favor of non-interlingual heterogeneous reasoning systems is that such systems allows modularity in proofs. We observe that it is possible to focus on one of the surface systems to draw necessary conclusions which might serve as milestones in the overall proof, and only when milestones are reached might we want to concern ourselves with information expressed in other systems. We can therefore separate different subtasks within a larger proof

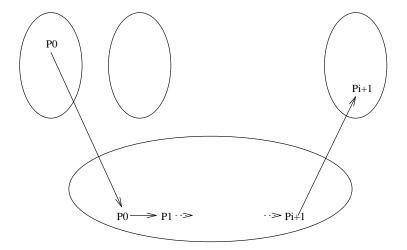


Figure 9: Interlingual Approach

by attending to information expressed in different systems at different times.

This efficiency is obtained by allowing control over the "exporting" of results from one system to another. As an example, we can imagine representing the initial hypotheses of a Hyperproof proof in the diagram, reasoning entirely within this representation, and then using the Observe rule to infer some new formulae. The proof may then proceed by inference within the sentential system, until a final result is obtained. This division of a single proof into sequences of inferences within a representation system with transitions between representations separating the sequences may be repeated many times within any individual proof.

There appear to be a couple of advantages in adopting a strategy like this in a heterogeneous reasoning system: there may be a psychological advantage, for a human being, in limiting attention to the consequences inferrable within a single representation system; and there may also be an advantage in being able to determine at various points in the proof which information will be made available and considered in the next part of the proof. However, whatever mechanism is used to determine when and whether to transition between representation systems could be used in an interlingual system to make the same decisions. For example, the interlingua may encode information with a "tag" indicating the surface system within which it was derived. The interlingual inference system could then give priority to inferences between information sharing a tag over information which does not. Whatever control strategy which is used to govern the search for a proof in the non-interlingual system will work just as well, in the interlingua, and this is true whether the cognitive agent is a human

being or a computer system. This argument is supported by the map reading example which we presented in section 5. Here, street and location information represented in different ways, although on the same map. Information in this interlingual representation may be tagged exactly as described; streets as red lines, significant locations as black dots, and perhaps contour information as thin black curves, and so on.

7.4 Custom Inference Rules

A final argument in favor of interlinguæ which seems appealing at first sight but which is questionable on closer examination is the idea that in surface representation systems we can make use of custom inference rules which are tailored to the individual representation in such a way as to make them easy for the agent to apply. This is true, of course, but the jump to believing that this advantage is necessarily lost when an interlingual system is adopted is questionable.

The map example which we cited in section 5, in which the surface systems are site locations in one system and street connectivity in the other, speaks to this feature of the system. There, we proposed the adoption of the combined map representation of this information as an interlingua. In this case the interlingua supports operators which are particularly suited for human use, as evidenced by the facility with which most of us can use maps. Thus, it seems that there is nothing about the intrinsic nature of an interlingua which necessarily entails that custom operators which facilitate relevant inferences cannot be used in an interlingua. This is obvious once it is recognized that an interlingua is just a representation system, like any other, which has a special role in an HRS.

8 Conclusion

We have done two things in this paper. First we defined formally the notion of heterogeneous reasoning system. We argued that an understanding of how we reason with information presented to us in a variety of notations is a necessary prerequisite to the design of effective multimedia applications for which the technological resources are becoming available. We presented some examples of heterogeneous reasoning systems in everyday use, and suggested that these might form the focus of an investigation of the general properties of such systems.

We also identified a number of interesting points in the space of possible heterogeneous reasoning systems. Systems with these more specialized properties seem to us to be worthy of significant investigation. Among these more specialized heterogeneous reasoning systems are the conversion HRS, in which heterogeneous inference may essentially be used simply to convert information from one representation to another, and interlingual heterogeneous reasoning systems, in which there is a distinguished representation capable of representing all of the information which is representable in the other surface systems. The notion of paradigmatically interlingual heterogeneous reasoning system seems to us to be particularly important, since these systems allow system designers to sidestep most of the problem of reasoning with heterogeneous information by translating it all into a common form. Objections have been raised to this approach, and the bulk of the second half of this paper is devoted to the examination of these objections.

We believe that in practice, when considering human beings reasoning with information presented in special-purpose representation systems designed for our use, the use of interlinguæ is probably inappropriate. On the other hand, we conclude that there is no a priori reason to suppose that this is true as a general matter. Just as the same logical properties can be implemented interlingually or non-interlingually, the same computational properties may also be available in the interlingual and non-interlingual cases. The main reason that interlingua seem not to be intuitively useful is that we have developed a wide range of surface representation systems which are particularly suited to the problems which we intend them to solve. The use of an interlingua for these representations is not likely to preserve these properties, and so it is often ill-advised to adopt this approach. However, this does not mean that the use of an interlingua is undesirable per se, but rather that given well-designed surface representation systems, like the examples cited by Barwise and Etchemendy, the use of an interlingua would not generally improve upon these systems.

We should discourage the use of interlingua only when we are confident that the representation systems which we are currently using have all the properties that we are likely to desire. Due to the evolving nature of notations, it seems unlikely that we will ever know that any particular notation that we have adopted is in fact optimal. In our map example above (section 5), we asked the user to imagine navigating with a map without place names marked, used in combination with a table of place, coordinate position pairs. All of the required information is available via such a joint representation, but this is not perceived to be as useful as a conventional map. After discovering the utility of notating the place names on the map, such a representation would not be classed as an interlingua for long. Since it is effective for human use, we would imagine that the new representation system would replace the original collection of surface systems, and itself become a surface system in general use. The promotion of an effective interlingua to a surface system may well be part of the evolution of notation for a particular domain. This claim seems to us to be amenable to historical confirmation.

References

- [1] G. Allwein and J. Barwise, editors. *Logical Reasoning with Diagrams*. Oxford University Press, 1996.
- [2] J. Barwise and J. Etchemendy. The Language of First-Order Logic (including Tarski's World). CSLI Press, 1992.
- [3] J. Barwise and J. Etchemendy. Hyperproof. CSLI Press, 1994. (ISBN: 1-881526-11-9).
- [4] J. Barwise and J. Etchemendy. Heterogeneous Logic. In J.I. Glasgow, N. H. Narayanan, and B. Chandrasekaran, editors, *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. AAAI Press/MIT Press, 1995.
- [5] K. D. Fisler. Exploiting the potential of diagrams in guiding hardware reasoning. In G. Allwein and Barwise J., editors, Logical Reasoning with Diagrams. Oxford University Press, 1996.
- [6] E. Hammer. Logic and Visual Information. Number 3 in Studies in Logic, Language and Information. CSLI Publications, 1995.
- [7] Sun-Joo Shin. *The Logical Status of Diagrams*. Cambridge University Press, 1995. (ISBN: 0-521-46157-X).
- [8] K. Stenning, R. Cox, and J. Oberlander. Contrasting the cognitive effects of graphical and sentential logic teaching: Reasoning, representation and individual differences. Language and Cognitive Processes, 10(3/4):333-354, 1995.