UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A 2b. DECLASSIFICATION / DOWNGRADING SCHEDULE N/A	3. DISTRIBUTION / AVAILABILITY OF REPORT Unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 40A	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Linguistic Research Institute 6b. OFFICE SYMBOL (If applicable) 5Y922	7a. NAME OF MONITORING ORGANIZATION RADC
6c. ADDRESS (City, State, and ZIP Code) 5600 Arapahoe Ave, Suite 206 Boulder, Colorado 80303	7b. ADDRESS (City, State, and ZIP Code) Griffiss AFB, New York 13441
8a. NAME OF FUNDING / SPONSORING ORGANIZATION RADC / IRDA 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F30602-85-C-0190
8c. ADDRESS (City, State, and ZIP Code) Griffiss AFB, New York 13441	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62702F PROJECT NO. 4594 TASK NO. 16 WORK UNIT ACCESSION NO. 12

11. TITLE (Include Security Classification)
Knowledge Base for C3I (Unclassified)

12. PERSONAL AUTHOR(S)
Peter G. Ossorio and Lowell S. Schneider

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Technical Rep.	FROM 9/85 TO 9/87	87 Oct 1	211

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Expert Systems, Artificial Intelligence, AI, Data Base, C3I, I&W

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This project investigated (a) the representational requirements and (b) the computer implementation requirements for applying the State of Affairs (SA) approach to a central type of I&W problem. The main knowledge base requirements were identified as tables of (a) social practice representations and (b) scenario representations, together with several auxiliary files. The main data base requirements were identified as (a) a fact file summarizing observations and (b) auxiliary files for statistical and SA facts derived from the fact file and the knowledge base. A prototypical computer implementation was developed.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL P. Langendorf	22b. TELEPHONE (Include Area Code) 315-330-3126 22c. OFFICE SYMBOL IRDW

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

KNOWLEDGE BASE FOR C3I

P.G. Ossorio and L.S. Schneider

October, 1987

Final Technical Report

F30602-85-C-0190

Submitted by
Linguistic Research Institute, Inc.
5600 Arapahoe Avenue
Boulder, Colorado 80303

Submitted to
Rome Air Development Center
Griffiss AFB, New York

EXECUTIVE SUMMARY

Artificial Intelligence aids for the central tasks of intelligence analysis appear to require a more powerful representational technology than is provided by currently well known AI methods.

State of Affairs methodology addresses directly the problem of representing real things (objects, processes, events, and states of affairs) in the real world.

It includes (a) a well grounded formal system, (b) a characteristic methodology, and (c) a representational technology for describing the elements, happenings, and situations encountered in the real world.

The Knowledge Base for C3I project was an investigation of (a) the representational requirements and (b) the computer implementation requirements for applying the State of Affairs approach to a central type of I&W problem.

The key representational requirement was found to be (a) a library of social practice and scenario representations and (b) an observation report data base with entries formatted to interact with the Social Practice and Scenario representations.

An analysis of functionality requirements for computer implementation was made. Prototype software for meeting the functionality requirements was developed.

Areas requiring further attention were surveyed. They include (a) the need for some additional complexities in the process representation, (b) the need to complete an inference engine capable of dealing with this level of complexity, and (c) the issues raised by designing for multiple users as contrasted with the single-user prototype.

No decisive problems were encountered in applying the SA approach to the I&W problem.

The relation of the SA approach to other AI approaches was examined. All were found to have less representational power than the SA approach and all would be redundant in areas covered by SA representation. Three approaches (Rules, Frames, Semantic Nets) are potentially complementary to SA; two others (Scripts, GOALS) are not.

TABLE OF CONTENTS

1.0 Introduction 6

2.0 The State of Affairs Formal System 8

3.0 State of Affairs Methodology 17

4.0 State of Affairs Representation 28

5.0 Representational Requirements 52

6.0 Choice Principle Analysis 83

7.0 Computer Implementation 90

8.0 Current Outlook 98

9.0 Relation to Other AI Approaches 105

10.0 Conclusions 115

APPENDICES

A. Installations and Facilities. 116

B. Forward Units 120

C. Path File 122

D. Preemptive Strike Scenario 127

E. Fact File (Data Base) 157

F. Softward User Manual 168

LIST OF TABLES

1. State of Affairs Transition Rules 9

2. Basic Process Unit (BPU) 30

3a. Basic Object Unit 41

3b. Extended BOU 47

4. State of Affairs Unit 50

5. Preemptive Strike Scenario Excerpt 55

6a. Social Practice Representation 61

6b. Task Analysis for SP 7.1 66

7. Data Base Excerpt 75

8. Fact Types 79

LIST OF FIGURES

1. Conventional Process Schema 32

2. Map of Domain 53

1.0 INTRODUCTION

1.1 State of Affairs Representation was developed to provide explicit, systematic representation of the objects, events, etc. in the real world. It offers a conceptual framework, a methodology, and a systematic notation. It is computer implementable.

1.2 State of Affairs Representation is particularly applicable to intelligence analysis because a State of Affairs representation provides the framework within which observational facts (including sensor-based facts) can be fitted together and systematically analyzed.

1.3 The Knowledge Base for C3I project was an investigation of the applicability of SA representation to intelligence analysis problems. Specific attention was given to the question of what actual real world representations would be needed to provide an effective framework for processing and integrating observational reports of various kinds. A second focus of attention was the analysis of observational data in terms of operative choice principles (values, attitudes, policies, strategies, doctrine). A third focus was the preliminary specification of a computer implementation.

1.4 In the following sections, we first present the State of Affairs system of representation. This is done in three parts, i.e., the basic conceptual system (Section 2.0), the characteristic methodology (Section 3.0), and the descriptive notation and technology for constructing representations of real world phenomena (Section 4.0). In subsequent sections, we apply these resources to the three tasks noted above, i.e., (a) what representations would be needed to provide an effective framework for processing and integrating observational reports of various kinds (Section 5.0); (b) the analysis of observational data in terms of choice principles such as attitudes, values, policies, strategies, and doctrine (Section 6.0), and (c) the specification of a computer implementation (Section 7.0).

2.0 THE STATE OF AFFAIRS FORMAL SYSTEM

2.1 In this section we present the State of Affairs approach as such. Although illustrations are selected with intelligence analysis in mind, the application of SA to intelligence analysis is dealt with in later sections.

2.2 The real world is what we see when we look around us. It is what we are each a part of and have a place in. It also includes the objects, events, and other phenomena that we act in the light of and in relation to. It includes, for example, other people, myself, automobiles, mountains, trees, military installations, military activities, children playing games, and the rain falling in the afternoon. It includes the activities of military intelligence and the facts that intelligence analysts have or would like to have.

2.3 The State of Affairs conceptual system is formalized as a set of Transition Rules, shown in Table 1.

2.3.1 Because the SA Transition Rules exemplify the Element-Operation-Product model of a formal system, the Transition Rules qualify as a formal system.

In such a system a finite set of Elements and Operations is explicitly introduced. By stipulation, Operations are performed on Elements (there may be

Table 1. State of Affairs Transition Rules

1. A state of affairs is a totality of related objects and/or processes and/or events and/or states of affairs.

2. A process (or object or event or state of affairs) is a state of affairs which is a constituent of some other state of affairs.

3. An object is a state of affairs having other, related objects as immediate constituents. (An object divides into related, smaller objects.)

4. A process is a sequential change from one state of affairs to another.

5. A process is a state of affairs having other, related processes as immediate constituents. (A process divides into related, sequential or parallel, smaller processes.)

6. An event is a direct change from one state of affairs to another.

7. An event is a state of affairs having two states of affairs (i.e., "before" and "after") as constituents.

8. That a given state of affairs has a given relationship (e.g., succession, incompatibility, inclusion, common constituents, etc.) to a second state of affairs is a state of affairs.

8a. That a given object or process or event has a given relationship to another object or process or event is a state of affairs.

9. That a given object, process, event, or state of affairs is of a given kind is a state of affairs.

10. That an object or process begins is an event and that it ends is a different event.

10a. That an object or process occurs (begins and ends) is a state of affairs having three states of affairs ("before," "during," and "after") as constituents.

restrictions on which Operations are allowed with which Elements). The result of performing an Operation on an Element is a Product. Every Product qualifies as an Element. Every Element can be represented simply as an Element or as an Element-Operation combination. For example in the arithmetic system, the same number can be represented simply as an Element (e.g., "12") or as any of a number of Element-Operation combinations (e.g., "4x3," "10+2," "26-14," "30/2.5," "(15+9)/2," ad infinitum).

2.3.1.1 To see that the Transition Rules exemplify this model, note that each rule consists of a left-hand element and a right-hand element connected by the word "is." These correspond, respectively, to Element, Product, and Operation. There is only one Operation in the system. That Operation is identity coordination (or simply, "identity"). The "is" in each Transition Rule is the "is" of identity; i.e., it can be paraphrased as "is the same thing as".

2.3.1.1.1 In the context of ordinary language discourse (as contrasted with theoretical, technical, or philosophical discourse) each rule tells us that what is described in the first way is the same thing as what is described in the second way. By implication, each rule tells us that there are two ways of describing the same thing. Since we are dealing with fundamental reality concepts, there is no third description that would tell us directly what it is that these two descriptions are descriptions of. Any third description would at best be merely a third description of the same thing, not the "real" description of it.

2.3.1.1.2 If this state of affairs seems to leave matters unbearably loose or up in the air, it may be helpful to recall that the description of motions presents us with exactly the same situation. We can give two or more descriptions of "the same motion" within different frameworks, but we have no privileged framework that would give us the "real" description of that motion. Yet we do very well with the phenomenon of motion, and we do so in practical, intuitive, observational, technical, and theoretical contexts.

2.3.2 The explicit primitives of the SA conceptual system are "Object", "Process", "Event", and "State of Affairs". ("Relation" may be regarded as a fifth primitive.)

2.3.2.1 Since these concepts are primitives in a formal system and (independently) since they are presented as fundamental concepts, it would obviously be inappropriate to expect definitions of "Object", "Process", "Event", and "State of Affairs". What we can expect, and what the

Transition Rules do provide, is (a) a set of paradigmatic sentences which show the logical grammar of each concept and which may be interpreted as (b) a set of general statements that tell us what is true of, e.g. objects, that is or is not also true of each of the others. (Note that although no one thing needs to be uniquely true of a given one of the four concepts, the set of things which are true of it must be unique, otherwise we have a redundant concept.)

2.3.2.2 If we consider the Transition Rules from this standpoint, we find that they do give us this kind of information. For example:

(a) Both process and events are happenings; they occur. In contrast, neither objects nor states of affairs are happenings. Objects merely are; they do not occur. States of affairs are, or are the case; they do not occur.

(b) A process has duration and involves successive happenings. An event has no duration and involves at most one succession. An object has duration but it does not as such involve any happenings. A state of affairs may or may not have duration or involve successive happenings.

(c) A process divides into related processes. Objects, events, and states of affairs can't divide into processes, though states of affairs can have processes as immediate constituents.

(d) An object divides into related objects. A process can't divide into objects. Nor can events.

(e) Both events and states of affairs divide into states of affairs; neither processes nor objects can divide into states of affairs.

2.3.2.3 The Transition Rule primitives are our most generic categories of what there is to be observed when we pay attention to what is around us.

The Transition Rules should be contrasted most pointedly with a <u>theory about</u> the real world. Such a theory would be couched in non-mundane technical terms and would, in effect, tell us that what we observe is really something else.

In contrast, it is not theorizing to say that blue is a color, that circles are round, or that colors are visible. Likewise, it is not theorizing to say (a) that a chair is an object, that <u>this</u> chair is an object, or (b) that the game being played by the children outside is a process, that troop movements or tank advances or airplane overflights are processes, or (c) that the beginning of the children's game was an event, that the airplane's reaching the border is an event, or (d) that the children being on vacation is a state of affairs, that there being only three tanks left to defend the fort is a state of affairs (and that the tanks and the fort are objects).

2.3.2.3.1 It is well established that an inventory of what there is in the world can be accomplished by using any one of four basic reality concepts. These concepts are "object," "process," "event," and "state of affairs."

2.3.2.3.1.1 In this context "real world" means "everything there is", not "the planet earth".

2.3.2.3.1.2 The world can be seen as consisting of a collection of objects of different sizes which exhibit a variety of relationships and interactions and together make up one huge object, namely, the world. Similarly, it can be seen as consisting of processes of various sorts which have various sorts of ingredients and together make up one all-encompassing process, namely the past, present and future history of the universe. And obviously, the world can be seen as consisting of an all-encompassing sequence of events or as an all-inclusive state of affairs.

2.3.2.3.1.3 The State of Affairs Transition Rules allow us to see (a) what is concealed by these single-category inventories of what there is, and (b) why it is that any of <u>these</u> four concepts will do the job.

The answer to the latter is obvious: Any of these four concepts (object, process, event, and state of affairs) will do the job because they are part of a conceptual system in which they are mutually defined and related so that any of them can substitute for the entire system.

What is concealed is that since each category is connected to the others, the existence of all is, in effect, guaranteed by the existence of any, and so we only have to assert the existence of one. Consider for example, the notion that the world consists of objects of different sizes which exhibit various relationships and interactions and together constitute the one huge object which is the world. So long as the component objects have relationships or are of particular kinds, there will be states of affairs (see Transition Rule 8a). So long as there are interactions among objects, there will be processes (See Rule 4). And so long as there are processes, there will be events (See Rule 10).

2.3.2.3.1.4 There may be some philosophical purpose in presenting single categories (object, process, event, state of affairs) as sufficient for an inventory of what there is. However, it is clear that for the purpose of representing what there is in the real world for scientific, technical, or practical purposes we shall need all four categories as well as the logical relations among them and the difference among them. These are provided by the SA Transition Rules.

In the context of intelligence analysis, we confront the problem not merely of correlating reports from different sources, but also of correlating reports of different kinds, e.g. reports of occurrences and happenings, reports of states of affairs, reports of possible happenings or states of affairs, etc. An explicit conceptual system for objects, processes, events, and states of affairs provides the potential for performing such correlations automatically.

3.0 STATE OF AFFAIRS METHODOLOGY

The State of Affairs conceptual system is part of the larger conceptual system of Descriptive Psychology. The other parts, which are comparable in scope, are conceptual formulations of Persons, Behavior, and Language. Implementation of this larger system, and, indeed, simply introducing the system has depended on methodological resources which have little in common with the traditional scientific reliance on mathematics, theories of logic, and cause-effect schemas. Instead, the primary methodology involves calculational systems, paradigm case formulations, and parametric analysis. Recall that the SA Transition Rules constitute a calculational system.

One of the major features of all three is that they serve to introduce a substantive domain without depending on anything external.

3.1 The notion of a calculational system as an Element-Operation-Product system was presented above in Section 2.3.1.

It should be noted that "Element" in this context is independent of the use of "Element" in the representational formats presented in Section 4.0 and used in later sections.

3.2 Parametric Analysis

A parametric analysis is a conceptual device for both (a) notationally identifying a range of cases as being a single subject matter, or domain, and (b) distinguishing among the cases.

3.2.1 Definition: To give a parametric analysis of a given domain of cases is to specify the ways in which one of those cases could be the same as another of those cases *as such* or different from it.

3.2.2 Note that such a specification will also permit us to specify the ways in which one *kind* of case within the domain could be the same as another kind of case within the domain or different from it.

3.2.3 Familiar examples of parametric analyses are the following.

(a) A traditional "fundamental analysis" in physics was a parametric analysis with Mass, Length, and Time as parameters. The claim was that all physical phenomena could be characterized *as* physical phenomena in these terms and that differences or similarities among various physical phenomena could be accounted for as differences or similarities in these three respects.

(b) The humanly visible range of colors can be effectively classified in a three-dimensional arrangement. Conventional names for the three dimensions are Brightness (the light-dark dimension), Hue (the rainbow color sequence), and Saturation (the gray-intense dimension).

3.2.4 Not every way of systematically classifying cases is a paradigmatic parametric analysis. Rather, a parametric analysis is a way of formulating the essential characteristics of the domain. (In this respect, a parametric analysis resembles a definition.) This is why the phrase "as such" is included in the definition of a parametric analysis and why it is underlined.

The contrast is with non-essential characteristics or empirical relationships with other things. For example, an automobile can be classified as a physical object. Automobiles can be classified in various ways, e.g., by the features that distinguish good-selling ones from unpopular ones or by the kind and amount of emissions, etc. These features are not essential ones. They are not what makes that object a physical object and they are not even what makes that object an automobile. Likewise, we can classify colors by how readily they are seen in a fog, by how pleasing or soothing they are to human observers, etc. These features are not what makes a color a color.

3.2.5 The descriptive formats presented in Section 4.0 reflect parametric analyses of the domain of Objects, the domain of Processes, and the domains of Events and States of Affairs. They are not arbitrary or merely instrumental ways of classifying Objects, Processes, etc. Rather they reflect an analysis of what is essential to the concept of an Object and of what is essential to the concept of a Process, etc.

3.2.6 Non-essential characteristics are provided for in the Transition Rules by Rules 8, 8a, and 9. (E.g., Rule 9: That a given object, process, event, or state of affairs is of a given kind is a state of affairs.) Only in the case of Objects is the possession of attributes explicitly mentioned in the basic formulation.

3.2.7 Each parameter in a parametric analysis has associated with it a set of values. Particular cases can be characterized not merely by the parameters, but more specifically by their parametric values. As a result, particular cases can be distinguished from other cases more or less precisely and more or less completely by specifying parametric values. This is the general thrust of the representational formats presented in Section 4.0.

3.3 Paradigm Case Formulation (PCF)

Like a parametric analysis, a paradigm case formulation is a formal device for both (a) identifying a range of cases as a single subject matter and (b) distinguishing among the cases. But whereas in a parametric analysis we simply analyze essential aspects of cases, in a paradigm case formulation we begin with a set of cases as such and then derive other cases.

3.3.1 A Paradigm Case Formulation consists of two parts. These are conventionally designated as Stage I and Stage II:

Stage I. Introduce a Paradigm Case

Stage II. Introduce some number of Transformations of the Paradigm Case

3.3.2 The mechanics of a PCF are as follows.

3.3.2.1 The specification of the Paradigm Case directly identifies a subset of cases in the domain. This specification must be directly understood or explainable in a finite number of steps in order to be useful to others (the same limitation that would hold if one were to give a definition).

3.3.2.2 Each Transformation will pick out additional cases. Each Transformation can be interpreted as an instruction: "Change the Paradigm Case in <u>this</u> way (the Transformation) and the result will still (also) be a case of X". It can also be interpreted as an indirect description: "If it (the Paradigm Case specification) were different in this way (the Transformation), it would still (nevertheless) be a specification of a set of cases of X".

3.3.2.3 Thus, if the PCF is successful, the Paradigm Case and the Transformations will, collectively, pick out all the cases of X and only cases of X.

3.3.3 A standard example of a Paradigm Case Formulation is the following PCF for the concept of a family.

I. Paradigm Case: A husband and his wife living with their natural children, who are a seventeen-year-old son, a ten-year-old daughter, and a five-year-old daughter.

II. Transformations:

T1. Eliminate one parent.

T2. Change the number of children to N, N > 0.

T3. Change the sex distribution of the children to any distribution other than zero boys and zero girls.

T4. Change the ages of the children to any values compatible with the ages of the parents.

T5. Add any number of additional parents.

T6. Add adopted and other legally defined sons and daughters.

T7. Eliminate the requirement of living together.

T8. Add zero children if husband and wife are living together.

T9. Eliminate the requirement that the parents have the legal status of married.

3.3.4 Like a Parametric Analysis, a PCF organizes a domain in terms of both similarities and differences. Unlike the Parametric Analysis, a PCF is particularly useful for distinguishing between essential and non-essential cases, or between central and peripheral cases, or between standard and non-standard cases, and so on. Where the range of cases contains two kinds which show this kind of contrast, an effective approach is to incorporate the standard (archetypal, etc.) cases in the Paradigm Case specification and introduce the non-standard cases via transformations.

3.3.4.1 An example can be found in the presentation, above, of a Parametric Analysis as dealing with the essentials of a concept or domain. Let us now consider that specification to be a Paradigm Case specification, and introduce the Transformation: T2. Eliminate the requirement that the parameters are the conceptual essentials. The result will be if only nominally, a Parametric Analysis.

3.3.4.2 This liberalization will extend the notion of a Parametric Analysis to almost any case of systematic classification. For example, the following would qualify.

(a) A tank mechanic's classification of tanks in terms of how they stand in regard to (their parametric values on) each of the major ways (the Parameters) that tanks can be dysfunctional.

(b) A marketing department's classification of automobiles (or furniture, etc.) in terms of the features that make them more or less desirable to the public.

(c) A "profile" which distinguishes between skyjackers and ordinary airplane passengers or between an Operational Maneuver Group and an ordinary Army unit.

3.4 A definition of "X" is the specification of the necessary and sufficient conditions for something to be a case of "X". Definitions involve primitives (undefined terms). So do theories, calculational systems, parametric analyses. There is no escape from the fact that explanations and classifications must come to an end at some point. Fortunately they are not always needed and therefore all of the formal devices noted above can be used successfully.

Since definitions and theories are the most conventional ways of introducing subject matter in scientific and technical activities, some comment on why Descriptive Psychology emphasizes the other three is appropriate. (Definitions and theories do have a place in Descriptive Psychology also.)

3.4.1 The brute fact is that satisfactory definitions are almost never achievable when it comes to anything in the real world, e.g. armies, airplanes, people, mountains, rivers, knowledge, motivation, consciousness, information, technology, science, etc., etc.

3.4.1.1 Another brute fact is that we do not acquire one of these concepts by virtue of having a definition and thereafter recognizing instances of it. Instead, we acquire a concept of this kind (a) by encountering a number of instances, and/or (b) by hearing (reading, etc.) how people talk about X's, and/or (c) by talking to people about X's, and/or (d) by observing how people treat cases of X, and/or (e) by how people talk about how people do, could, should (etc.) treat cases of X.

3.4.1.2 Nor is it the case that it appears to us that there really are definitions for all those things, but we just haven't achieved them yet. On the contrary, it usually seems quite clear that the difficulties with definitions stem from the fact that there isn't a set of necessary and sufficient conditions for "X", that there is nothing that all and only cases of "X" have in common that make them cases of "X". For example, looking at the Paradigm Case Formulation for families above, it seems clear that there is nothing that all families have in common that is essential to their being families; conversely, it is equally clear that the collection of cases given by the PCF is conceptually coherent, and not just an arbitrary collection.

In short it appears that real world phenomena are conceptual primitives which we can classify in a variety of ways which serve various purposes but cannot define except approximately and for limited purposes.

3.4.1.3 Exhibit A (From the American Heritage Dictionary, New College Edition)

Object: (1) Anything perceptible by one or more of the senses, especially something that can be seen and felt; a material thing

Process: (1) A system of operations in the production of something

(2) A series of actions, changes, or functions that bring about an end or result

(3) Ongoing movement; progression

Event: (1) An occurrence, incident, or experience, especially one of some significance

(2) The actual outcome or final result

Fact (an approximation to "State of Affairs"):

(1) Something known with certainty

(2) Something asserted as certain

(3) Something that has been objectively verified

(4) Something having real, demonstrable existence

3.4.2 To talk about real objects, events, etc. in terms of either definitions or theories involves the introduction of a new set of terms and concepts. In short, it involves talking about something other than the objects, events, etc. we wanted to talk about. Given the difficulties with definitions of real world phenomena and the scarcity of useful theories concerning them, definitional and theoretical approaches are hardly impressive resources for dealing with such matters.

Calculational systems, Parametric Analyses, and Paradigm Case Formulations free us substantively from the major methodological hazards of definitions and theories. They allow us to deal with real world phenomena as primitives and they allow us to talk about real world phenomena <u>as</u> real world phenomena.

4.0 STATE OF AFFAIRS REPRESENTATION

The world consists of some collection of objects, processes, events, and state of affairs related to one another in the ways given by the Transition Rules. But of course, that cannot be the whole story. Rather, the world consists of particular objects of particular kinds, particular processes of particular kinds, and particular events and states of affairs of particular kinds, related to one another in particular ways. The Transition Rules only set the broadest limits on what might be the case and on what must or cannot be the case; they do not tell us what there in fact is in the world or what is in fact the case.

In most areas of science and technology it is axiomatic that we have to find out about the world primarily by observation and secondarily by thought. If we accept this axiom, then our efforts to find out about the world will be directed toward (a) making observations of whatever is of interest, (b) developing models which, in conjunction with observation help us anticipate phenomena of interest (but never in a foolproof way) and (c) achieving a single coherent representation of all the things we found out about separately. State of Affairs representation is explicitly designed for the third of these and makes at least two significant and distinctive contributions to the second.

By using parametric analysis methodology on the concepts of "object," "process," "event," and "state of affairs," we arrive at the logical schemas needed to distinguish one object (or process, etc.) from another or, correspondingly, to give a systematic description of a particular object (or process, etc.) or kind of object (etc.). These are presented below.

4.1 Process Representation and Process Description

Table 2 shows the Basic Process Unit, which is the essential logical schema for representing particular process or types of process.

4.1.1 The first thing to note is that the Basic Process Unit (BPU) has a two-part structure. The two parts are (a) Name and (b) Description. The first allows us to represent a process merely as a single entity; the second allows us to represent a process as an articulated whole with an internal structure. For a technology of representation, the latter is of primary interest.

4.1.2 The first specification, i.e. "Stages" reflects Transition Rule 4, i.e. "A process is a sequential change from one state of affairs to another." This implies that for a given process, there is not only an initial state of

Table 2. Basic Process Unit (BPU)

P-NameA: The process "Name" of process A.

P-DescriptionA: The "Description" of A. It specifies:

I. P-Paradigms: The major varieties of P-NameA. This is a technical option. If only one paradigm exists, it will be the same as P-NameA. For each paradigm, the following is specified:

(a) Stages I-K: These are "Names" of subprocesses within A. They are systematically specified, e.g., as P-NameA11, P-NameA12,...,P-Name A1K for Paradigm 1. For each stage, specify:

(1) Options 1-N: These are the various exemplars of the process (stage) in question. That is, these are the various ways in which that process could happen. Each Option is systematically indexed as P-NameA111,P-NameA112,..., P-NameA11N. Each of these can now be expanded (decomposed) on the model of P-NameA.

(b) Individuals

(c) Elements

(d) Eligibilities

(e) Contingencies

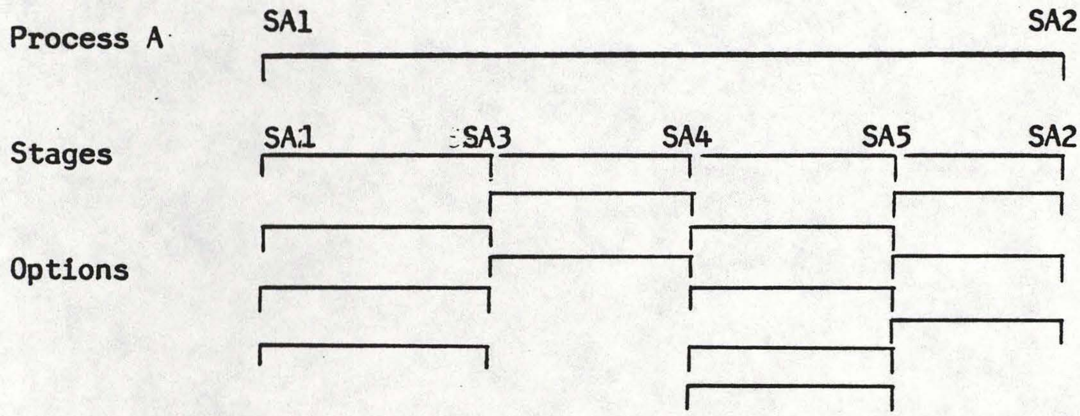
(f) Versions

affairs and a final state of affairs, but also at least one intermediate state of affairs (otherwise the process would not be a sequential change). Such an intermediate state of affairs would, by Rule 5, divide the original process into a sequence of two sub-processes. Sub-processes of this kind are designated as "stages" of the process. These general features of processes are shown graphically in Figure 1, which is a conventional schema for showing a process as having a Stage/Option structure.

4.1.3 In general, a process can happen in more than one way, and this is shown explicitly in regard to the Stages. Options are relative to Stages. An Option for a given Stage is simply one of the ways that that sub-process can happen.

4.1.4 Since an Option for a given stage is a process, it too can be represented by means of the BPU format. Likewise, the original process which was divided into Stages may itself be represented as a stage in a larger process which in turn can be represented by means of the BPU format. In this way, representations can be "composed" into larger representations and they can be "decomposed" into representations of smaller components. This allows us to represent phenomena of different magnitudes within the same framework.

Figure 1. Conventional Process Schema



4.1.5 The Stage/Option Structure is only the gross structure of a process. Which Stages a process has (and how many) and which Options a Stage has (and how many) are genuinely ways in which one process can, as a process, be the same as another process or different from it. However, that is only the beginning.

4.1.6 Another parameter is Elements. Elements are the formal "ingredients" of the process. For most processes, all or most of the Elements will correspond to objects. For example, if one thinks of a drama as a paradigm case of a process, the various characters in the drama, along with all the required material objects will qualify as the Elements in that process. More formally, the initial state of affairs for the process (SA1 in Figure 1) is (by Rule 1) a totality of related objects (etc.). The Elements are these formal objects (etc.). The relationships among these objects change as the process proceeds and these changes correspond to intermediate states of affairs (e.g., SA3, SA4, SA5 in Figure 1). The sequential changes in these relationships is (is the same thing as) the occurrence of the process.

4.1.7 In the simplest case, when an instance of the process takes place, each Element is embodied by a separate historical individual. For example, if the drama is Hamlet, then John Doe plays Hamlet and nothing else; Richard Jones plays Polonius and nothing else; only one object serves as the skull in the graveyard scene, and so on. Formal Individual and Eligibility specifications allow for more complex possibilities, where the same Individual is eligible to play the part of more than one Element. Thus, in an actual occurrence of the process, there is a one-to-one relationship between historical individuals and formal Individuals, rather than between historical individuals and Elements.

4.1.7.1 Elements and Individuals/Eligibilities are ways in which one process can, as a process, be the same as another process or different from it. In general, such specifications allow us to represent that the particulars of the occurrence of a process (including its outcome) depend on the ingredients on any given occasion (including what would normally be called "initial conditions"). In this way, the logical schema for process representation embodies the truism that "the same process" can have different outcomes, depending on initial conditions.

4.1.8 The occurrence of a process on a given occasion is the same thing as the occurrence of one of the Options for Stage 1, followed by one of the Options for Stage 2, followed by one of the Options for Stage 3, and so on. In the simplest case, the occurrence of the process could consist of any of the Options for Stage 1 followed by _any_ of the Options for Stage 2 followed by _any_ of the options for Stage 3, and so on. In fact, most processes, particularly social and behavioral processes are more complex than that. The Contingencies parameter allows us to represent such complexity. Contingency specifications are ways of restricting the availability of Options in a given process by making the availability contingent on something. We distinguish four sorts of Contingencies, i.e., Co-occurrence, Attributional, Relational, and Factual.

4.1.8.1 Co-occurrence contingencies specify that a given Option in a given state is available (for instantiating the process) if and only if certain other Options in other stages are the ones selected (or are not selected). For this aspect of process description a game with simple sequential process structure, e.g., chess, provides the most obvious example. In chess, clearly, the Options chosen on the first move make a crucial difference in regard to which Options are available on the second move, and both are crucial in regard to which Options are available on the third move, and so on.

4.1.8.2 Attributional Contingencies specify that certain Options are available (for the occurring of the process) only if the individuals playing the parts of given Elements have certain (specified) attributes. In this way we can represent processes in which certain possibilities depend on (usually exceptional) characteristics of the individuals involved.

The range of examples of such cases includes the following.

(a) An 80-yard pass is one of the Options in a football game _if_ the offensive team is behind their 20 yard line and the quarterback has an extraordinarily strong arm. (b) In a skirmish, shooting an enemy through a ten-inch tree trunk is not an option if the ammunition is a .30 caliber copper jacket. A tank has the option of moving directly across the river if it is an amphibious tank. If we move from certainties to likelihoods, other sorts of examples arise. For example, a chess player has the option of play the Kings Gambit _if_ he believes his opponent cannot play it well. A commander who is trying to achieve maximum impact will be likely to move surreptitiously at first if there is an opportunity.

In general, the kinds of human attributes that enter into Contingency specifications are attitudes, traits, values, knowledge or beliefs, abilities, policies, strategies, and doctrine.

4.1.8.3 Relationship Contingencies specify that a given Option is available only if the individual playing the part of a given Element has a given relationship or relationships to given individuals, organizations, objects, resources, and so on. Relationship contingencies and attributional contingencies are completely analogous.

4.1.8.4 Factual Contingencies specify that a given Option is available only under certain conditions (e.g., only if the weather is good or if lakes and rivers are frozen solid). Any restriction on the availability of Options which cannot be stated as a co-occurrence, attributional, or relational contingency can be stated as a Factual Contingency.

4.1.9 Specifying States, Options, Elements, Individuals/Eligibilities, and Contingencies provides the restrictions needed to specify what would qualify as an instance of the process in question. Any sequence that would qualify as an instance is a Version of the process. The occurrence of a process is the same as the occurrence of one of its Versions.

4.1.10 The specification of process Paradigms is a technical convenience. It is appropriate when the Versions of a process divide into two or more subsets with distinctive Elements or structures. For example eleven man football vs. seven man football, shaving with an electric razor, a straight razor, and a safety razor, lectures vs. field trips, a direct river crossing vs. a crossing on temporary bridges. In such cases, the representation is much more clear and useful if it is segregated into two or more process paradigms each with a standard process representation.

4.1.11 An important special case of a process representation is the Social Practice representation. In the latter, the ultimate constituent processes are the behaviors of individual persons. This is the archetypal case. In practice it need not hold without exception. For example, the Social Practice representation in Table 6a (Section 5.0) below contains one Stage which is not a behavior.

4.1.12 The actual occurrence of a process is a historically particular happening. It requires that there be appropriate historical individuals to play the part of each formal Individual and therefore, each Element. The concept of a process is the concept of an abstract structure of proceedings, ingredients, events, and states of affairs, and any actual collection which exemplifies that structure is an occurrence of the process.

Several familiar examples provide heuristic models for the various aspects of processes. (a) A chess game is a prime model for the Stage/Option structure of a process and for co-occurrence contingencies. (b) A drama such as Hamlet is an excellent model for the Element/Individual/Eligibility aspects of a process. (c) A recipe is also an excellent model for the Element/Individual/Eligibility aspects and for both attributional and co-occurrence Contingencies, as well as the Stage structure. Note that as a prescription for the process of making a particular sort of food, a recipe would need to make the potentially problematic aspects of the process relatively explicit and complete.

4.1.13 A process representation is the representation of the abstract structure of a certain kind of process as such (e.g., a chess game). A process description is the description of something as an instance of a certain kind of process (e.g., "this is a chess game"). Where complex processes are involved, we first construct a complex process representation and then we can say of an actual process, "This is one of these."

4.1.14 A process representation need not be complete in order to be useful. It is generally feasible to represent processes only down to a certain level of detail beyond which the particulars don't matter. In addition to incompleteness in terms of detail, there are two systematically incomplete sorts of process representation or process description.

4.1.14.1 A means-ends description refers to Elements of a process (the "means") and the outcome of that process (the "ends"). What is left out is the Stage/Option and Contingency structure of the process. This kind of description is useful, for example, in dealing with resource allocation issues or in capability analysis.

4.1.14.2 Achievement analysis (or task analysis) refers only to the outcome aspects of a process. There are two kinds of analysis. In the first kind, the final outcome is analyzed into components such that if each of the components is separately achieved, the final outcome is achieved. In the second kind, the achievement of the final outcome is analyzed into a sequence of achievements which, if actually achieved, would bring about the final outcome.

4.2 Object Representation and Object Description

A "Basic Object Unit" (BOU) for representing objects is developed in accordance with the Transition rules. Table 3a shows the BOU.

Table 3a. Basic Object Unit

O-NameA: An expression which identifies the object. (This may be expanded to a list of names, each of which is the name of this same object.)

O-DescriptionA: The "description" of O-NameA. It specifies: O-Paradigms 1,2,...n. These are alternate decompositions of O-NameA into immediate constituents. For each paradigm, specify:

(1) Constituents: A list of immediate consituents, which for systematic purposes may be designated as O-Name1A1, O-Name1A2, ..., for paradigm 1; O-Name2A1, O-Name2A2, ..., for paradigm 2, etc. (In practice such "descriptive" names as "carburetor," "hand," "pancreas," will also be used.) Each such constituent can now be decomposed by being given a BOU representation.

(2) Relationships 1, 2,..., m: These are given by a list of relationships. Each item on the list is specified as follows.

(A) Name: An expression which identifies an N-Place relationship (state of affairs). Note that n is not constant for different elements of the list 1, 2,..., m.

(B) Elements: A list of N Elements, each of which is one of the members of the N-place relationships.

(C) Individuals: A list of individuals which are constituents of O-Name1A.

(D) Eligibilities: A specification of which individuals may or must participate as which Elements in the relationship by virtue of their constituency in O-Name1A.

(E) Contingencies (Attributional or co-occurrence): Specification of conditions under which an individual eligible to be a given Element is that Element.

(3) Attributes of O-Paradigm (i.e., of O-NameA as consisting of the structure given by the relationship involved in a given paradigm).

4.2.1 The most relevant Transition rules are the following.

(a) An object is a state of affairs having other, related objects as constituents.

(b) A state of affairs is a totality of related objects and/or processes and/or events and/or state of affairs.

4.2.2 Like the BPU, the BOU has a Name/Description structure. This allows us either to give a systematic description or to do no more than refer to the Object by using the Name.

4.2.3 An object divides into smaller, related objects which are its parts. In general, there are indefinitely many ways to divide an object into immediate constituents. (The immediate constituents of an object are those (a) into which it is directly divided (as against being the result of repeated divisions) and (b) which jointly make up the object. Therefore the Basic Object Unit provides for an indefinite number of alternative ways of dividing an object. Each of these is represented as an Object Paradigm (O-Paradigm).

4.2.3.1 A complex object such as an automobile, a tree, a computer, or a human body will in general have a finite number of obvious and non-arbitrary ways of dividing it. The number of viable alternatives will be reduced by the uses for which the object representation is designed. Homogeneous objects such as a marble or a mound of earth, do not offer obvious ways of dividing them into a structure of parts, but conversely, there is much less likelihood that any systematic description beyond a Name would be needed.

4.2.4 It follows straightforwardly that for each Object Paradigm the major specifications are of the constituents and the relationships among them.

4.2.4.1 The immediate constituents are given by a list. Although only one name for each is needed, computer implementation would generally be facilitated by having a set of names including, for example, (a) a systematic name which would identify the original object, the Object Paradigm, and the number (on the list) of each constituent object, (b) one or more descriptive names that identify which constituent it is, e.g. "hand," "carburetor," "leaf," and (c) one or more names which identify it as an independent object and do not presuppose that it is part of the original object.

4.2.4.2 The relationships are also given by a list, with each entry on the list corresponding to one relationship. For each entry, the relationship specification has the following components.

(a) First, there is a name which designated an N-Place relationship. (Note that when we divide an object into K parts, there are 2-place relationships between every two parts, 3-place relationships among every three parts, etc.)

(b) Second, there is a set of N Elements, each of which is one of the terms (arguments, members) in the N-Place relationship.

(c) Third, there is a list of Individuals each of which is eligible to play the part of one or more of the N Elements.

(d) Finally, there are Contingencies (most notably co-occurrence or attributional contingencies) which specify the conditions under which an Individual eligible to be a given Element is that Element. (Note that the parts of an object must match each other, and the more strongly structured the object is, the stronger the restrictions on the match; for example an object may be a perfectly good firing pin, but not for <u>this</u> weapon.)

4.2.4.3 The attributes of the Object Paradigm, i.e. the object considered as <u>that</u> structure of <u>those</u> components, is given by a list.

4.2.4.4 One reason for the complex specification of relationships among constituents is that object representation must allow for cases where those relationships may change (in certain ways) without changing the object into something else. Objects may have processes going on involving their parts.

4.2.4.5 As with process descriptions, object descriptions need not be literally exhaustive in order to be useful or in order to be exhaustive for practical purposes. Two sorts of object description which are systematically incomplete are of interest.

(a) One can specify only the name and the attributes (either directly or via the specification of a value on an attribute dimension). In this case, the object is dealt with as a unitary entity without regard to structure.

(b) Once can specify only the name and the constituents without specifying relationships. In this case, the object is dealt with as a collection of constituents.

4.2.4.6 Object representations are suitable for representing stable structures of any kind, not merely material objects. In particular, organizations and social systems can be represented as objects whose constituents have certain required social relationships (role relationships, status relationships).

4.2.4.7 Table 3b shows the Extended BOU. This makes explicit some of the complexities which would be involved in a multilevel object representation. Of particular interest is contingency 2C. Most technical terminology, including military technical terminology, falls under this contingency. For example to speak of "the Third Army" or of "Colonel Johnson" is to speak of an organization or an individual not as an independent (context-free) thing but rather as components, and particular components or types of component at that.

Table 3b. Extended BOU

For O-NameA, specify:

(1) Attributes of O-NameA

(2) Contingencies:

(A) Attributes which a given constituent must have.

(B) Applicability of a given name, e.g., O-Name2A3, as presupposing a given O-paradigm or a recursive elaboration of one.

(C) Applicability of a given name as presupposing a unit (e.g., object, process, configuration) of which O-NameA is a constituent. (Most technical terminology would fall under this heading -- recall "the physical world," "the baseball world," etc.)

(D) Configuration (etc.) membership or attributes of O-NameA as contingent on the specification of a given O-Paradigm, KA, or

(E) on given attributes of a given constituent (e.g., an automobile is an internal combustion machine because its motor is an internal combustion machine.)

(3) Relationships: As in Table 3a, above, but not restricted to immediate constituents as Elements.

(4) (Optional convenience) Configuration membership: A list of configurations of which O-NameA may be a constituent.

4.2.4.8 Most objects are routinely referred to in terms of their place in larger structures or processes. For example, machines, tools, and other sorts of equipment are routinely referred to in terms of their paradigmatic uses even when they are not or cannot be used in that way. For example, a Leopard tank being used as a research test bed is still called a tank and it would still be called that if it was obsolete and inoperable. Likewise a baseball bat is called that almost under any conditions and not just when it is being used as a baseball bat. Similarly for missiles, aircraft, chess sets, skis, colonels, chimneys, bridges, intelligence analysts, eyes, chairs, and tails. (The phenomenon has the general structure of a paradigm case formulation, i.e. reference is anchored by a paradigmatic use or place and is retained over a range of transformations.)

4.2.4.9 The fact that, in both ordinary language and technical language, objects are routinely referred to as (what amounts to) Elements of either social practices or social groups has at least one significant consequence. That is that observation reports can in general be expected to be reports about Elements. In turn this suggests that a system with a knowledge base composed of social practice representations and object representations has the potential for effectively accepting and interpreting observation reports with a minimum of preprocessing.

4.3 State of Affairs Representation

Table 4 shows the State of Affairs Unit (SAU). As with the Basic Process Unit and the Basic Object Unit it has a Name/Description structure and an Element/Individual/Eligibility structure in the Description.

4.3.1 The basic specification for SA representation is that of an N-place relation and the terms of that relation (the Elements).

4.3.2 Because a state of affairs is a totality of related objects and/or processes and/or events and/or states of affairs, the Elements are specified as to which of the four they are or may be. Likewise the Individuals are so classified. A set of Individuals which meet the category specifications and the requirements of the specified relationships will constitute an instance of the state of affairs.

Table 4. State of Affairs Unit

SA-NameA: The "Name" of state of affairs A. This may be given by any identifying reference, such as a sentence ("The man shot the bear"), a sentential clause ("the shooting of the bear"), a simpler locution ("the shooting"), or a conventional symbol (SA-NameA).

SA-DescriptionA: The "Description" of SA-NameA. It specifies:

(I) <u>Relationship:</u> An explicit identification of an N-place relationship, or attribute. (A property is a 1-place attribute.)

(II) <u>Elements:</u> A list of the N elements, or logical roles in the Relationship. These are distinguished as 1st, 2nd,...,Nth elements.

(IIa) <u>Eligibilities:</u> Each of the N elements is characterized as being either necessary or optionally an object, process, event, state of affairs, attribute, or concept.

(III) <u>Individuals:</u> A list of N Individuals identified <u>as</u> individuals by a name, number, symbol, etc. (Note that "individual" is <u>not</u> the same as "object".)

(IIIa) <u>Classification:</u> Each of the N individuals is identified as an object, process, event, state of affairs, attribute, or concept.

(IV) <u>Assignments:</u> The N Individuals are placed in one-to-one relation with the N Elements, with each Individual being identified as <u>the</u> exemplar of the corresponding Element in <u>the</u> state of affairs SA-NameA.

(V) <u>Expansions:</u> An expansion consists of the recursive use of the SAU (as developed to this point) in one of the following ways:

(1) Elaborating the Classification of a given Individual as an object, process, event, or state of affairs by giving a SAU description of it (via Rule 1,3,5, or 6). This will amount to using BPU, BOU, Event, or SAU formats.

Table 4. State of Affairs Unit (Con'd)

(2) Elaborating the Classification of a given Individual as an Attribute by giving a SAU description in which the Attribute is the Relationship.

(VI) <u>Contingencies:</u>

(1) Since contingency statements are possible within BPU and BOU representations and the latter may occur as expansions, such contingency statements will qualify as contingencies within the full SAU also.

(2) Co-occurrence constraints such that the use of a particular "Name" (in general, referring terminology, either technical or nontechnical) for any Element within the full SAU is contingent on the use of particular other "Names" for other Elements.

(3) Co-occurrence constraints such that the use of a particular Element is contingent on its being that element (or an Element) of the SAU within which it is an Element. (Note that stages, options, and paradigms within a BPU or BOU will qualify as Elements here.)

5.0 REPRESENTATIONAL REQUIREMENTS

The problem addressed was that of determining the representational requirements for the application of SA System representation to the tasks of intelligence analysis. The general area of I&W was chosen as an appropriate starting point.

5.1 In principle, I&W deals with any factor or phenomena which indicate the increased likelihood of aggressive action in areas of interest. In practice, most indicators have to do with the preparatory activities or early stages of aggressive action.

5.2 The initial problem was to set the stage in a realistic fashion for the technical problem of representation. The two initial decisions were to select a geographic area as the domain of interest and to define a prototypical aggressive action as the substantive issue.

5.2.1 The geographic area chosen was the western portion of Czechoslovakia and the southern border region of East Germany, roughly, the area from Prague to Eisenach at the western border of the DDR. This area is shown in Figure 2. This choice was made primarily on the basis of a heuristic OB type data base covering this region. The data base was

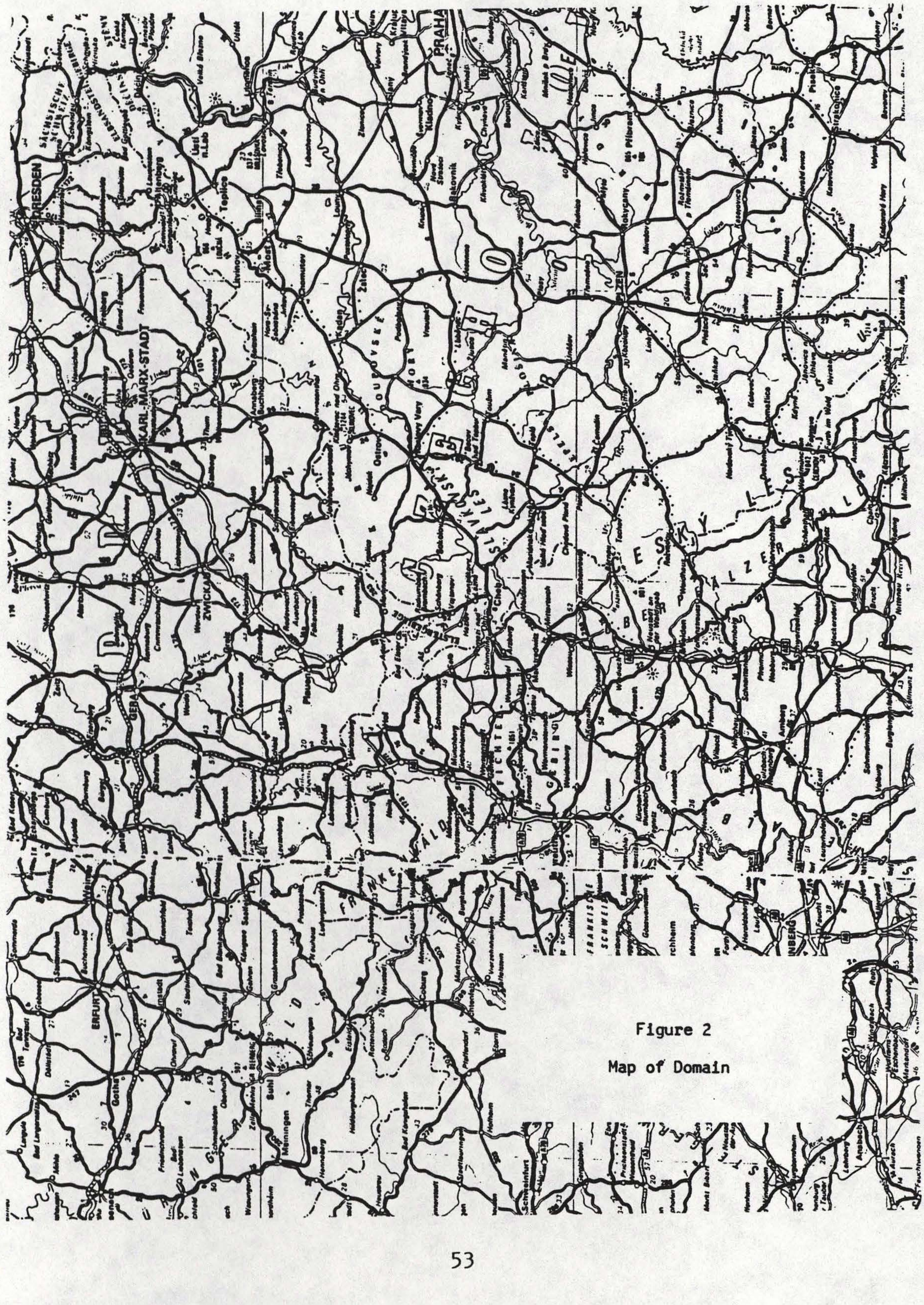


Figure 2

Map of Domain

made available through the courtesy of Knowledge Systems Concepts, Rome, N.Y.

5.2.2 The prototype aggressive action was a scenario involving an attack across the border initially disguised as a training exercise. The scenario was constructed with specific reference to the geographic region, the road system, and the immediate targets. This was designated as the "Preemptive Strike Scenario" (PS Scenario).

The PS Scenario is shown in part in Table 5 and in its entirety in Appendix D. Although it conforms roughly to the BPU format, the emphasis was on the content rather than on rigorous representation.

Table 5. Preemptive Strike Scenario (Excerpt)

1: Announce Large Pact Exercise involving ground and air forces

1.1: Option 1: Live Fire Exercise

1.1.1: Stage 1: Use Diplomatic Channels to announce exercise

1.1.2: Stage 2: Call Major Commanders from Germany & Poland

1.1.3: Stage 3: Issue Misleading Exercise oriented messages

1.2: Option 2: Mere Exercise

1.2.1: Stage 1: Call commanders for briefing on "exercise"

1.2.2: Stage 2: Announce exercise locally

1.2.3: Stage 3: Misleading troop movements locally - rehearsal

1.3: Option 3: Command and Control Exercise

1.3.1: Stage 1: Brief Commanders

1.3.2: Stage 2: Conceal Troop Movements by spreading over time

2: Build up resource reserves for operations

2.1: Increase production at Prag munitions plant

2.1.1: Stage 1: Increase rate of arrival of raw materials

2.1.1.1: Option 1: Conceal by holding constant # but increase load

2.1.1.2: Option 2: Increase both # of shipments and loads

2.1.2: Stage 2: Increase the employment and shifts at plants

5.2.3 Domain Representation

In principle, the application of SA representation to I&W requires a general representation of the domain of interest, since this is the context within which the phenomena which are observed and reported all have an intelligible place.

5.2.3.1 Formally, the obvious choice for an overall domain representation is an Object representation. That would include the representation of component objects and the activities and interactions of these components. In general, the major types of components would be the following:

(a) The geographical terrain itself, with significant terrain features differentiated as component objects.

(b) Stationary installations and facilities such as bridges, highways, factories, railway yards and tracks, and various military installations and facilities.

(c) Mobile individuals, including individual persons; organizations, including troop units; and mobile or portable equipment, weapons, tools, etc.

5.2.3.2 The following types of installations and facilities were included in the domain description (a complete listing of actual cases is given in Appendix A).

(a) Bridges
(b) Power Facilities
(c) Railroad Facilities
(d) Military Facilities
(e) Repair and Maintenance Facilities
(f) Munitions Storage Depots
(g) Weapons Facilities
(h) Air Bases
(i) POL Storage Facilities
(j) Radar and Fire Control Facilities
(k) Missile Sites

5.2.3.3 In addition, the domain description included a Ground OB, with particular attention to the forward units (Appendix B) and an analysis of the paths between points of interest (Appendix C). The types of path analyzed include the following:

(a) From refinery or shipping point to POL storage areas

(b) From refinery, shipping point, or POL storage areas to air bases

(c) From munitions factory to forward units at Greiz, Suhl, and Plzen

(d) From forward units to tank repair and maintenance yards

5.2.3.3.1 The path analysis includes the following.

(a) Point of origin
(b) Destination
(c) Sequence of highways
(d) Direction of travel on each highway
(e) Estimated normal travel time.

5.2.3.3.2 The path analysis facilitates the interpretation of certain observation reports. For example, the report of a convoy of 12 POL Tanker Trucks observed on Highway E63 is interpretable as

(a) 12 Tanker Trucks on the path from the KMS refinery to the Jena air base

(b) 12 Tanker Trucks on the path from the KMS refinery to the Blankernhain air base

(c) 12 Tanker Trucks on the path from the KMS refinery to the Waltershausen air base

5.3 Social Practice Representation

Given the representation of the various paths, installations, and facilities and their locations relative to the terrain, the next problem was to provide a representation for possible happenings (including the component states of affairs) which would be of interest. The indicated form of representation is process representation, and specifically, social practice representation.

5.3.1 The choice of social practices for representation came primarily from the PS Scenario. Since a Scenario is defined as (1) an ad hoc structure of social practices (2) with some specification of historical (i.e., actual) individuals, the PS Scenario was analyzed into a set of social practices. Representation of these practices would make possible the recognition of the occurrence of instances of them, including the instances compatible with the PS Scenario. For present purposes, implementation of a sample of these practices (indicated in Appendix D) was judged to be sufficient. The selection emphasized the early stages of the scenario, since the detection of the early stages of an activity has the biggest I&W payoff. For a more extensive project, there would be a variety of different scenarios and systematic representation of all the social practices involved.

5.3.2 For each social practice representation a corresponding Achievement Analysis (Task Analysis) was performed. The Achievement Analysis is derived from the social practice representation by specifying (a) the overall outcome (state of affairs) of the occurrence of the social practice and (b) for each stage, the state of affairs which marks the end of that stage and the initial condition for the next.

5.3.2.1 The outcomes in this case are the logical context-free outcomes, not empirical, context-dependent ones. For example, if a stage in a process is "[Tanker Truck] delivers [POL] to [POL Storage Facility]" the logical outcome is "Truckload of [POL] has been delivered to [POL Storage Facility]." (Brackets indicate process Elements.) Identification of context-dependent empirical outcomes (e.g., the storage tank is full; this is the tenth truckload this week; the POL delivered will keep the unit operating for ten days) depend on having a representation of the context; this possibility is provided by the domain representation, including observed facts (see Section 5.4, below).

5.3.2.2 Achievement Analyses can be performed directly in the absence of the corresponding process representation. They provide an alternative vehicle for identifying happenings in the domain.

5.3.3 An example of a social practice representation which exemplifies the Basic Process Unit format is shown below in Table 6a and the corresponding Achievement Analysis is shown in Table 6b. It should be noted that bracketed terms (e.g. [TTruck]) are Element names. The names were selected for their mnemonic or heuristic value in identifying the part played by that Element in the process as a whole. For computer processing, they could be replaced by "Element 1," "Element 2," etc.

Table 6a. SOCIAL PRACTICE REPRESENTATION

SP 7.0 [Vehicle] delivers [Material] at [Site]

P-Paradigm 7.1 [TTruck] delivers [POL] at [POL Storage Site]

(Stage 1)	7.1.1	[Driver] moves [TTruck] into [Delivery Position]
(Stage 2)	7.1.2	[Driver] dismounts from [TTruck]
(Stage 3)	7.1.3	[Driver] connects [EConnector] to [SConnector]
(Stage 4)	7.1.4	[Driver] activates [Delivery Control]
(Stage 5)	7.1.5	[POL] flows from [TTruck] to [Storage Structure]
(Stage 6)	7.1.6	[Driver] checks completeness of delivery
(Option 1)	7.1.6.1	[Driver] observes [Gauge]
(Option 2)	7.1.6.2	[Driver] observes flow of [POL]
(Stage 7)	7.1.7	[Driver] deactivates [Delivery Control]
(Stage 8)	7.1.8	[Driver] disconnects [EConnector] from [SConnector]
(Stage 9)	7.1.9	[Driver] restores [EConnector] to original position
(Stage 10)	7.1.10	[Driver] mounts [TTruck]
(Stage 11)	7.1.11	[Driver] moves [TTruck] to [Readiness Position]

Table 6a. SOCIAL PRACTICE REPRESENTATION (Con'd)

SP 7.0 [Vehicle] delivers [Material] at [Site]

P-Paradigm 7.1 [TTruck] delivers [POL] at [POL Storage Site]

Elements:

[TTruck] =	Tank Truck or any vehicle capable of carrying bulk quantities of [POL]
[POL] =	Any petroleum-derived fuel or lubricant
[POL Storage Site] =	Any separate facility for storing [POL]
[Driver] =	The driver of the [TTruck]
[Delivery Position] =	[TTruck] position in which transfer of [POL] takes place
[EConnector] =	Equipment which connects tank on truck to storage structure
[SConnector] =	Equipment on storage structure which connects to truck
[Delivery Control] =	Equipment which directly controls the delivery of [POL]
[Readiness Position] =	Position which completes delivery and from which next action can be taken
[Gauge] =	Any metering device for delivery of [POL]
[Storage Structure] =	Structure in which [POL] is stored

Table 6a. SOCIAL PRACTICE REPRESENTATION (Con'd)

SP 7.0 [Vehicle] delivers [Material] at [Site]

P-Paradigm 7.1 [TTruck] delivers [POL] at [POL Storage Site]

Individuals/Eligibilities:

1. [POL Storage Site]
   a. Bor POL Storage Site
   b. Schneeberg POL Storage Facility
   c. KMS POL Complex
   d. Schleiz Bulk POL Storage Facility
   e. Neustadt Jet-A Storage and Pumping Station
   f. Grafenrod POL Storage Area
   g. Bad Salzungen POL Storage Site
2. [TTruck]
   a. Any tank truck
   b. Any POL vehicle
3. [Driver]
   a. // A non-com or enlisted man
   b. A civilian
4. [EConnector]
   a. // A hose
   b. A spigot
5. [SConnector]
   a. A hose/sleeve connection

Table 6a. SOCIAL PRACTICE REPRESENTATION (Con'd)

SP 7.0 [Vehicle] delivers [Material] at [Site]

P-Paradigm 7.1 [TTruck] delivers [POL] at [POL Storage Site]

b. An aperture in underground storage or delivery structure

c. A sleeve connection

6. [Delivery control]

a. A valve

b. A pump

7. [POL]

a. POL

b. BOL

c. MoGas

8. [Readiness Position]

a. A location at or near [POL Storage Site]

b. A designated location at or near [POL Storage Site]

9. [Delivery Position]

a. A location adjacent to storage stucture

b. A location adjacent to delivery point other than storage structure

10. [Gauge]

a. Metering device with dial

b. A level indicator

Table 6a. SOCIAL PRACTICE REPRESENTATION (Con'd)

SP 7.0 [Vehicle] delivers [Material] at [Site]

P-Paradigm 7.1 [TTruck] delivers [POL] at [POL Storage Site]

11. [Storage Structure]
    a. Above ground tank
    b. Underground tank
    c. Bladder

Contingencies:

(Co-Occ) 1. [Driver] is same individual in all stages

(Co-Occ) 2. [TTruck] is same individual in all stages

(Co-Occ) 3. [POL Storage Site] is same individual in all stages

(Co-Occ) 4. [POL] is same material in all stages

(Co-Occ) 5. [EConnector] is same individual in all stages

(Co-Occ) 6. [SConnector] is same individual in all stages

(Co-Occ) 7. [Delivery Control] is same individual in all stages

(Attrib) 8. [Driver] is a person

(Relation) 9. [EConnector] is part of [TTruck]

(Relation) 10. [SConnector] is part of [Storage Structure]

(Relation) 11. [Gauge] is part of [TTruck] or part of [Storage Structure]

(Fact) 12. Stage 1 iff [EConnector] and [SConnector] can be connected

Table 6b. TASK ANALYSIS FOR SP 7.1

SP 7.0 [Vehicle] delivers [Material] at [Site]

P-Paradigm 7.1 [TTruck] delivers [POL] at [POL Storage Site]

(Initial Condition)	0.	[TTruck] has load of [POL]
		[TTruck] is at [POL Storage Site]
(Stage 1 Achievement)	1.	[TTruck] is in position to deliver [POL]
(Stage 2 Achievement)	2.	[Driver] is dismounted
		[Driver] is in position to make delivery
(Stage 3 Achievement)	3.	[TTruck] is ready for delivery
(Stage 4 Achievement)	4.	[POL] delivery begins
(Stage 5 Achievement)	5.	[POL] delivery proceeds
(Stage 6 Achievement)	6.	[POL] is delivered
(Stage 7 Achievement)	7.	Delivery mechanism is stopped
(Stage 8 Achievement)	8.	[EConnector] is disconnected from [SConnector]
(Stage 9 Achievement)	9.	[EConnector] is restored to original position
		[TTruck] is restored to traveling condition
(Stage 10 Achievement)	10.	[Driver] is in position to drive [TTruck]
(Stage 11 Achievement)	11.	[TTruck] is ready for next action
(Overall Achievement)	12.	[TTruck] has delivered [POL] at [POL Storage Site]

The examples in Tables 6a and 6b bring to light a number of technical issues.

5.3.3.1 In general, phenomena should be represented at "too detailed" a level.

An examination of the social practices in Table 6a in light of the preemptive strike scenario will show that what is of interest is the occurence or non-occurence of these practices rather than the individual behaviors which are their components and which provide most of the content of the social practice representations.

This is a general feature of SA systems. In the hierarchical representation, it is desirable to represent not only the phenomena of interest, but also the next level of detail and perhaps even the next further level of detail.

This is particularly true for phenomena which have many components (e.g. staging a major mission) and/or which develop slowly and/or covertly over time (e.g., the buildup of personnel, equipment, and material for a major offensive).

The reason is that in such cases it is unlikely that the occurence of the phenomenon of interest would be established (in timely fashion, as against long after the fact) by a simple report or observation. Rather, reports and observations would most likely refer to some of the component activities, objects, facts, etc., and timely conclusions about the phenomenon of interest would have to be based on these.

One example may have particular interest for intelligence analysis: In the case where a set of aggressive activities is being carried out under a cover of normalcy, there is some chance that the "normal" covering activities will not be carried out in full detail for reasons of practicality. Having a detailed representation of the normal practices would facilitate the detection of deception.

5.3.3.2 Social practices may have other social practices as constituents.

For example, the social practice of Table 6a (i.e., SP 7.0: [Vehicle] delivers [Material] at [Site]) is part of another social practice, namely, SP 8.0: [Convoy] delivers [Material] at [Site]. Likewise, the P-Paradigm actually shown in Table 6a (i.e. SP 7.1: [TTruck] delivers [POL] at [POL Storage Site]) is a part of another P-Paradigm, namely, SP 8.1: [TTruck Convoy] delivers [POL] to [POL Storage Site].

5.3.3.2.1 The existence of hierarchical part-whole relationships among social practices points up the practical necessity for conventions dealing with naming and identity.

Ideally, every Option, every Stage, and every Social Practice should be given by an independent, stand alone representation. However, following this principle literally leads to a mass of trivial co-occurrence contingencies such as contingencies 1-7 in the social practice representation in Table 6a. At present we have introduced a convention that assumes than an individual playing the part of a given Element is the same individual across all stages of a Social Practice, and exceptions will be codified in contingencies or additional Options.

5.3.3.2.2 This solution is not available across Social Practice level boundaries, particularly when we have generic practices and more specific process paradigms. This relationship is exemplified in Table 6a, where the generic SP is 7.0: [Vehicle] delivers [Material] at [Site] and the particular process paradigm is 7.1: [TTruck] delivers [POL] at [POL Storage Site]. In this case, management of the instantiation relation could be accomplished by means of contingency specifications of the form "7.1 iff [Vehicle] = [TTruck]."

5.3.3.3 The example in Table 6a also exemplifies the tradeoff between increasing the number of paradigms and increasing the number of contingencies. Paradigm 7.1 could be paralleled by another paradigm, 7.2 representing the case where [POL Storage Site] personnel do the actual delivery rather than the truck driver. In that case, 7.2 would read the same as 7.1 except that in Stages 2-10, [Driver] would be replaced by [SPersonnel]. If both were combined into one representation, there would be two Options in each of these stages. For example:

(Stage 4) 7.1.4 [Delivery Control] is activated

(Option 1) 7.1.4.1 [Driver] activates [Delivery Control]

(Option 2) 7.1.4.2 [SPersonnel] activates [Delivery Control]

This form of representation would require a set of co-occurence Contingency specifications linking the [Driver] Options and another set linking the [SPersonnel] Options. For example:

```
7.1.4.1   IFF  7.1.3.1
7.1.6.1   IFF  7.1.4.1
7.1.7.1   IFF  7.1.6.1
etc.
7.1.4.2   IFF  7.1.3.2
7.1.6.2   IFF  7.1.4.2
7.1.7.2   IFF  7.1.6.2
etc.
```

From these examples, it is clear that simplicity and effectiveness of representation depends on suitable choices in the form of representation.

5.3.3.4 The Basic Process Unit has formal Individuals eligible to play the part of Elements and then, for an actual occurrence, historical (actual) individuals play the part of the formal Individuals and by virtue of that, play the part of Elements. In the representation of SP 7.1 the formal Individuals are bypassed and the Eligibility table directly connects historical individuals to Elements. For example, the Bor POL Storage Area is a historical individual which is eligible to play the part of [POL Storage Site] in SP 7.1.

The possibility of implementing this simplification stems from the choice of a Domain and the availability of the static Domain representation. It follows from the latter that there is a finite set of historical individuals capable of playing the part of [POL Storage Site] and presumably that set does not change rapidly. This would be true no matter how large a geographic area was chosen from the Domain.

5.3.3.5 Not all Options or individual-Element instantiations are on a par. Some are more or less paradigmatic, by virtue of being archetypal, normative, typical, or customary, and others are not. It is useful, therefore to have what amount to "normality" labels for

Options and instantiations, either in categorical or numerical form. The advantages include:

(a) Quick searches using only the paradigmatic Options, and

(b) More ready detection of unusual happenings. To achieve effective labeling of this sort adds significantly to the representational task, so that the question of feasibility remains open.

Table 6a illustrates the use of a categorical label i.e., a double slash ("//") to indicate paradigmatic instantiations:

3. [Driver]

    (a) // A non-com or enlisted man

    (b) A civilian

4. [EConnector]

    (a) // A hose

    (b) A spigot

5.3.3.6 In a sense, the notion of doing quick searches using only the paradigmatic Options is already involved in some of the simplifications noted above. For, whereas the conceptual framework and notation are geared to encompassing all the possibilities, in practice it is always necessary to stop somewhere short of that, based on the judgment that <u>these</u> are the possibilities we know about and care enough about to represent them.

5.3.3.7 SP 7.1 shows preparation and recovery as part of the delivery process. Other possibilities would be to make them separate practices and/or to make them part of a larger, contextual practice. In the case of SP 7.1, the larger practice is SP 8.0: [Convoy] delivers [Material] at [Site]. The choice was made on the basis that inclusion of preparation and recovery as part of the practice makes the practice a self-contained unit and therefore facilitates a modular approach to processing SP representations.

5.3.3.8 Both SP 7.1 and the corresponding Task Analysis are shown in Table 6b. Potentially, the use of Task Analyses rather than the full SP representations is a cost effective simplification. Since the Task Analysis is simpler, it provides fewer observational connections to what is going on, but, correspondingly, it is less complicated to process.

5.4 The elements of domain representation dealt with so far, i.e. the installations and facilities, highways and paths, and locations and composition of forward units provide a picture of what is in the domain. They do not provide a picture of what is happening in the domain.

A dynamic representation of the domain is achieved by means of a data base incorporating observational data together with several kinds of interpretive processing.

5.4.1 Table 7 shows an excerpt of the data base developed for the present project. The types of information contained in the data base reflect a process of

(a) Stipulating a number of routine activities occurring in the domain, and

(b) estimating the kinds of observation which could plausibly be expected to be made.

The "routine activities" also reflect the kinds of action involved in the PS Scenario.

Note that the core data in each entry is a relation or activity involving up to four Elements. To this is added (1) a Quantity/Units qualifier, (2) a Date/Time label, (3) an observer credibility label, and (4) a Fact Type label.

5.4.1.1 The data in the Fact File shown in Table 7 are not an arbitrary or random collection of facts. Rather, they represent an implicit simulation, since they are some of the things that might be observed if a stipulated history of activities and happenings were the same as the actual history of the domain.

Since the data in the Fact File reflects the assumption that whatever was in principle observable or discoverable by an outside observer might in fact be directly reported, the facts are of various kinds and some of them might be of interest in their own right. In general, however, the facts

Table 7. Data Base Excerpt (Left half)

Fact Type	Action	Element	Element	Element
17	Deliver	TTrucks	Karlovy Vary Air Base	POL
4	Depart	TTrucks	Manenberg POL Facility	*
4	Depart	TTrucks	KMS POL Facility	*
16	Deliver	TTrucks	POL	Neustadt POL Facility
1	Depart	TATrucks	Suhl Barracks	Tanks
5	On	TTrucks	Hiway E63	*
5	On	TTrucks	Hiway 174	*
5	On	TTrucks	Hiway 19	*
14	Parked at	Tanks	Plzen Tank Yard	*
17	Deliver	TTrucks	POL	Dobris Air Base
4	Depart	TTrucks	Manenberg POL Facility	*
4	Depart	TTrucks	KMS POL Facility	*
5	On	TTrucks	Hiway 169	*
5	On	TTrucks	Hiway 12	*
14	Parked at	Tanks	Gera Tank Yard	*

Table 7. Data Base Excerpt (Right half)

Element	Quantity	Units	Date	Time	Cred
*	15000	Gal	01-21-88	*	3
*	10	Ea	01-21-88	0900	2
*	13	Ea	01-21-88	0715	2
*	20000	Gal	01-21-88	*	3
*	6	Ea	01-21-88	0815	2
*	13	Ea	01-21-88	1110	2
*	10	Ea	01-21-88	0740	2
*	13	Ea	01-21-88	0930	2
*	12	Ea	01-21-88	1420	2
*	12000	Gal	01-22-88	*	3
*	8	Ea	01-22-88	0715	2
*	14	Ea	01-22-88	0910	2
*	14	Ea	01-22-88	0945	2
*	8	Ea	01-22-88	1100	2
*	11	Ea	01-22-88	1540	2

are fairly atomic. The Fact File, therefore exemplifies the "Hard Data" problem, i.e., that the data is only informative in the context of other data and even then only after some kind of analysis.

5.4.2 The social practice representations provide a major resource for "interpreting" the "hard facts" in the data base.

5.4.2.1 The key operation is matching a given fact in the data base with one of the facts involved in the occurrence of a given social practice. Essentially one moves from the report that certain states of affairs were observed to the conclusion that a Version of a given social practice has taken place when (a) enough of the observational data matches the components of the social practice, (b) the data does not match any alternative practice as well as it does this one, and (c) there is no conflicting data. Failing this, what is available is a periodically updated log of which practices are compatible with the data, and for each one, how compatible in terms of degree of match.

5.4.2.2 The matching process is facilitated by the fact that observation reports are generally couched in terms of Elements rather than individuals or they make reference to individuals which are easily classified as Elements. For example, in "Twelve Tank Trucks departed from the Schneeberg POL facility at 1345 1/15," "Tank Trucks" is an Element name and "Schneeberg POL facility" is readily identified as [POL Storage Facility] by virtue of the Eligibility lists associated with process representations (see Table 6a for an example involving the Schneeberg POL facility).

5.4.2.3 Nevertheless, simple correspondence between Fact File data and Stage/Option/Practice descriptions or Achievement descriptions is not something that can be counted on in general. Two methodologies for facilitating the matching are plausible. They are not mutually exclusive.

5.4.2.3.1 The first is the familiar thesaurus method. In this case, each of the lines in the process representation would have a list of "synonymous" expressions and a given data item would match that part of the process if it matched any of the synonyms. The major drawback to be anticipated is that the greater complexity might lead to problems of economy or efficiency.

5.4.2.3.2 The second method makes use of the "Fact Type" label in column one of the data base entry. If the same system of labeling were used on each line of a process representation or Task Analysis then direct matching on the basis of Fact Type would once more be feasible. Table 8 shows an excerpt from the list of Fact Types developed to date.

Note that matching on the basis of Fact Type is a near-equivalent to matching on the basis of Activity and Elements, since Fact Types refer to both.

Table 8. Fact Types

Type	Description
1	Flatbed trucks depart from munitions factory
2	Flatbed trucks on highway
3	Flatbed trucks arrive at military facility ammo depot
4	Tank trucks depart from POL refinery or shipping point
5	Tank trucks on highway
6	Tank trucks arrive at POL storage facility
7	Tank trucks arrive at air base POL storage facility
8	Tank trucks at POL storage facility
9	Tank trucks at air base
10	Tank carriers depart from army compound
11	Tank carriers on highway
12	Tank carriers arrive at tank yard
13	Tank carriers at tank yard
14	Tanks parked at tank yard
15	Tank trucks depart from POL Storage Facility

5.4.2.4 Since scenarios are structures having social practices as components, the ability to interpret the database information as evidence that a certain social practice took place or is taking place will translate into the ability to use the database information as evidence that a certain scenario is taking place or did take place. Scenario detection requires both the detection of the component social practices and matching the historical individuals specified in the scenario representation. For example, if the scenario calls for "Troops deploy along Highway 62", then it is not enough to detect an instance of "[Force] deploys along [Highway]: one must also establish that the actual highway involved is Highway 62.

5.4.2.5 Matching can be done "in reverse" to good effect also. If the data match two or more alternative social practices, an examination of the fact types which differentiate the social practices would lead to a specification of what observations would need to be made in order to be confident of which one had taken place or was taking place.

5.4.3 Another major resource for interpreting the "hard facts" in the Fact File is the ability to perform statistical and other analyses, including trend analyses on the data. Part of the value of representing routine activities is to provide benchmarks of various sorts so as to facilitate the detection of unusual activity.

5.4.3.1 Evidence of unusual activities together with an inconclusive partial match of the data to a scenario of interest would almost certainly be a more powerful indicator of the occurrence of the scenario than either piece of evidence separately.

5.4.3.2 The structure of the Fact File makes it possible (in a computer implementation -- see Section 7.0) for an individual to define an ad hoc data base containing only specified types of data in order to facilitate individually defined analyses or updated display packages, etc.

5.4.4 The results of analyses described above constitute an additional set of (non-observational) facts.

5.4.4.1 The results of statistical analyses can be represented in a "Summaries File".

5.4.4.2 The results of matching to process representations can be represented in an "Interpretation File" having the same form as the Fact File.

5.4.4.3 The results of choice principle analysis can be represented in a "Individual Characteristics File".

5.5 Summary of Representational Requirements

The representational requirements described above are summarized as follows:

1. Domain Representation

1.1 Stable Domain Representation

Installation/Facility File

OB Files; TOE Files

Path File

Individual Characteristics File

Fact Type File

1.2 Dynamic Domain Representation

Fact File

Summaries File

Interpretations File

2. Process Representation

Social Practice Representations

Task Analyses

Scenario Representations

Alternatively:

1. Knowledge Base

Stable Domain Representation

Process Representation

2. Data Base

Fact File

Summaries File

Interpretations File

6.0 CHOICE PRINCIPLE ANALYSIS

"It's not what you do -- it's how you do it." This truism is, in effect, codified by the notions of Co-occurrence Contingency and Relationship Contingency. The first of these specifies that a certain option is available (for the occurrence of the Process) only if the individual playing the relevant Element has certain attributes. The second is parallel but specifies that the individual has a specified relationship.

Any person characteristic (attribute) may make a difference in how one does what one is doing. Among the kinds of characteristics which are of most interest for intelligence analysis are traits, attitudes, values, knowledge, abilities, policies, strategies, states of mind, and doctrine. These appear to be the characteristics which are most likely to make significant differences in relevant activities.

6.1 Co-occurrence contingencies and relationship contingencies were presented in Section 4 as logical constraints. Using the PCF methodology we can extend the formulation to less-than-certain connections, e.g., probabilities, degrees of confidence, etc. Thus, we can say that the availability of a given Option is increased or decreased relative to baseline level (corresponding to what a hypothetical "average person" could be expected to do). One reason why such connections are less than certain is that any individual has various person characteristics which operate with various degrees of priority; in general any given person characteristic is able to find expression in more than one Option in a given Social Practice, and any given Option will appeal differentially to more than one Person Characteristic.

6.2 A classic example of the connection between Option choices and Person characteristics is provided by the heuristic image of "Dinner at 8:30":

Wil: You know, I finished work at 6:00 yesterday and got home at 6:30. We had dinner at 8:30 and it was steak, well done.

Gil: So what else is new? In this yuppy town half the population could say pretty much the same thing.

Wil: Well, you know, yesterday morning I had a big argument with my wife and I left in the middle of it. I usually do get home at 6:30 but we usually have dinner at 7:30, not 8:30. I see you're smiling. And I like steak a great deal, but I like it rare -- I hate it well done.

Gil: She must have really been angry at you. She was really giving you the business.

6.3 In this example it is clear that (a) Wil's wife was serving dinner (the social practice) in one of the ways it can be done and (b) there was more to it than that -- the choice of just <u>those</u> options (8:30, steak well done) was an expression of hostility. Indeed, it wasn't what she did, but how she did it.

6.3.1 The example illustrates a fundamental fact, namely that the connection between Option choices and Person Characteristics is a two-way street.

(a) Person Characteristics have a selective influence on Option choices. Person characteristics are expressed by Option choices. Because of this,

(b) Option choices are the paradigmatic basis on which Person Characteristics are assessed. When a person consistently makes choices which express the same Person Characteristic we have greater confidence

(1) that the person does have the characteristic and

(2) that the characteristic is a stable one.

6.4 If a person has a stable person characteristic we have some confidence that he will continue to have it for some time to come and will continue to express it in some of his choices of Options in future activities, particularly (a) those Options which are the best expressions of the person characteristic and/or (b) those Options where other considerations are least likely to influence the choice.

6.5 All of the foregoing will, with the relevant modifications, apply to organizations, groups, political units, all of which can be meaningfully said to "do" things, and to do one thing rather than another (that could have been done) on a given occasion and over repeated occasions.

6.5.1 The choices and characteristics of organizations, etc. often are a relatively direct reflection of the choices and characteristics of the person or persons in leadership positions.

6.6 The representation of social behavior (including military activities) as embodying social practices with a Stage/Option structure and Attributional and Relational Contingencies provides the basic ingredients for choice principle analysis. The primary payoff is having an additional resource for correctly anticipating the behavior of the other person or group and additional clues as to how to prevent, encourage, or otherwise deal effectively with it.

6.7 What is needed for choice principle analysis, in addition to the basics provided by the Social Practice Representation, is to make explicit a set of connections between Options and Attributes of persons or groups.

6.7.1 The simplest procedure would be to use Attributes to categorize Options, and in the simplest case an Option would fall under one and only one Attribute. Thus, for example, Option 9.1.4.3 (Option 3 of Stage 4 of Paradigm 1 of SP 9) would be classified as an "Aggressive" Option and Options 9.4.1.1 and 6.3.2.2 might be "Conservative" Options, and so on. The Options chosen for this treatment would be simply those Options for which such categorization was plausible.

6.7.1.1 An examination of the Preemptive Strike Scenario shows the systematic inclusion of concealment and non-concealment Options. Concealment Options in turn can be interpreted as Hostile, or Aggressive, Options. If the Options are present in the Scenario, they will also be present in the corresponding Social Practice representations and, as described above, occurrences of these Options can be detected on the basis of the observational data in the Fact File.

6.7.1.2 Note that we <u>are</u> dealing with less than certain phenomena. Even in the Dinner at 8:30 example, we would say that Gil's conclusion was obvious but not that it was necessarily true or that it followed from the facts reported by Wil. Nor does it appear that we could set an empirically justified probability level or quantitative confidence level here. What we can say is that <u>at face value it looks that way</u>; that is why, and that is the sense in which, it is obvious.

6.7.1.3 Thus, the problem of categorizing Options is not a narrowly empirical one, but rather the problem of judging the face value of choosing that Option as against other Options.

6.7.1.4 In turn, the face value of choosing a given Option can be judged in a completely context-free way or with any degree of contextual specifications. The Dinner at 8:30 scenario illustrates the difference that contextual specification can make, i.e., the difference between "So what else is new?..." and "She must have really been angry at you..." In contrast, the "Aggressive" interpretation of concealment Options in the PS Scenario is relatively context free.

6.7.1.5 The tradeoff is that the more context is specified, the more informative the choice analysis can be expected to be but also, the more its usefulness is restricted to just the specified context. Presumably, cost/benefit considerations prohibit a choice principle analysis for every context. It would be possible, however, to give generic context specifications which would provide some of the benefits of context specification without unduly restricting the range of application of the choice principle analysis.

6.7.2 Instead of simple categorization of Options under Attributes it is possible to supply numerical indices for Object-Attribute connections. In this approach, an Option is characterized by a profile which relates it quantitatively to each Attribute of interest. (An Attribute in turn can be characterized by a profile which relates it quantitatively to each of the Options which were judged.)

6.7.2.1 Issues of context specification noted above apply to the quantitative approach as well as the categorical approach.

6.7.2.2 There is a well developed methodology for implementing the quantitative approach. It is the judgment space methodology initially developed for automatic indexing and retrieval (Ossorio, 1964). Thus, there does not appear to be any substantial uncertainty as to whether choice principle analysis could be carried out for operational purposes in the context of the kind of domain representation and social practice representation described above in Section 5.0.

7.0 COMPUTER IMPLEMENTATION

One of the aims of the present project was to give preliminary specifications for computer implementation and, where feasible, to embody the specifications in prototype software.

7.1 Functionality

The first concern was to lay out the central functionality requirements for a computer implementation. To a large extent these follow from the analysis of the representational requirements presented in Section 5.0.

7.1.1 The representational requirements were summarized as comprising the following files.

(a) Knowledge Base:

Social Practice Representations (SP File)

Task Analyses (SP File)

Scenario Analyses (SP File)

Installation/Facility File

OB Files

TOE Files

Path File

Fact Type File

Individual Characteristics File

(b) Data Base:

Fact File

Summaries File

Interpretations File

7.1.2 The primary functional requirement is clearly the capability for matching the facts in the SP File with facts in the Fact File.

7.1.2.1 This involves several auxiliary requirements.

(a) Transforming data from the "present on highway X" form in which they appear in the Fact File to the "present on the path from A to B" form in which they can be matched to SP File data.

(b) Accessing OB Files and TOE Files to construct paradigmatic Object descriptions for, e.g., military units. A unit, e.g., the 17th Motorized Rifle Division, will be assumed to consist of the personnel and equipment called for by the TOE unless there is data in the Fact File to indicate otherwise.

(c) Enforcing restrictions imposed by Contingency and Eligibility specifications. Since these specifications will appear as tables (functionally) within the process representations, it appears likely that implementation can be achieved by means of Merge and Sort operations in a Relational DBMS. (However, see 7.1.2.2. This is an area that calls for further analysis.)

7.1.2.2 Although any single SP-Fact matching operation can be accomplished by a Relational DBMS, the systematic matching of the various Options in the various Stages in the various Social Practices in the various Scenarios calls for an inference engine with forward and backward chaining capability.

7.1.2.3 Given the basic matching capability, there is also the requirement for establishing the degree of match at the level of Social Practice, Scenario, or Task Analysis and not merely at the Stage/Option level. This calculation would be based on a User-selected set of parameters which might include any of the following.

(a) Number of possible matches (if the Social Practice had 16 Options in its various Stages, then the number of possible matches would be 16 or less, depending on whether some Options were mutually exclusive)

(b) Number of matches

(c) Proportion of matches (Note that 10/20 would generally be better than 1/2)

(d) Credibility of matching Fact File data

(e) Degree of ambiguity in matching the sequence of Stages (other matching refers to the occurrence of Options)

(f) Inconsistency of the occurrence of the Social Practice with other existing data

7.1.2.4 The foregoing capabilities would implement the following kinds of User query.

(a) Is Process X (or Scenario X) happening?

How compatible are the data with that?

(b) Given the facts, what processes are happening?

What alternative sets of processes are compatible with the data? How compatible are they?

What observational facts would distinguish among these possibilities?

7.1.3 If the choice principle analysis took the form of a Judgment Space, the following types of query could be implemented.

(a) For Colonel Sergei Rachmaninoff, show me the profile of the extent to which his decision making expresses each of the choice principles in the Judgment Space.

(b) Assume Colonel S.R. is in charge of Force X in Scenario SC 8.0. List the different Versions of the scenario in the order of likelihood based only on this assumption (or on this and specific other assumptions.)

(c) (In a situation where two or more processes are compatible with the data in the Fact File, and using the type of calculation used in (b), above) If Colonel S.R. is in command of Force X, does the likelihood of the Process A or Process B change? If so, how?

7.1.4 Additional functionality requirements have to do with performing a core set of statistical analyses on the data in the Fact File and the Individual Characteristics File and a core set of second order statistical analyses on the data in the Summaries File and Interpretations File. An alternative to the latter would be to perform trend analyses by systematically doing first order analyses within successive time slices. Thus, the analyses would include the following.

(a) Selection of type of data for analyses

(b) Means, sigmas for selected data sets

(c) Moving averages

(d) Time series

(e) Correlations

(f) Specific relations or comparisons, e.g. response time

In addition, for flexibility, data files should be available to User-specific applications programs.

7.1.4.1 The major uses of these analyses would be:

(a) To detect trends in the "normal" activities in the domain.

(b) To detect unusual happenings, events, or states of affairs within the domain.

(c) To provide data (either of the above) which could be integrated with the process interpretations. (For example, marginal conclusions concerning the occurrence of the PS Scenario would be strengthened by evidence of certain trends or unusual events. Recall that in the original discursive form (Appendix D) the scenario refers to "increases" of various sorts; these are not process facts except at a very high level of description -- they are more directly states of affairs concerning differences between states of affairs [specifically, some Achievements] at different times.)

7.1.4.2 Statistical analyses of the sort mentioned above are well understood and commonly implemented. Thus, no special problems are anticipated concerning them except insofar as they add complexity to the system as a whole.

7.1.5 Finally, there are various Input-Output issues. Among those which can be anticipated are the following.

(a) On the input side, observational data requires pre-processing in order to provide entries in the Fact File.

(b) The system must be able to access other data bases, since the relevant data cannot be expected to be in a single place.

(c) On the output side, the system should be able to print and display any of the files, any selected data set, and the results of any given analysis.

7.2 Implementation

Some of the functionality has been implemented in prototype software. The implementation has been guided by two assumptions.

7.2.1 The first is that a multi-user system is ultimately a requirement, but not an immediate one. The current implementation is a single-user system with a user interface which allows the user considerable latitude in creating, manipulating, and displaying data and data files.

7.2.2 The second is that since (a) relational data base systems are a relatively well understood technique and (b) much of knowledge base files and data base files specified for the system clearly lend themselves to this kind of treatment, it would be advantageous to try to operate as much as possible with a relational data base system. The alternative is to develop a custom architecture specifically responsive to the representational requirements and the functionality requirements.

7.2.3 A User's Manual for the relational data system and user interface is presented in Appendix F. This manual describes in some detail the functional resources provided by the software.

7.2.4 All of the software development to data has been directed toward (a) the use of the process representations to interpret facts in the Fact File and (b) the user interface for querying, displaying files, and creating and manipulating files.

7.2.5 Within this scope, the major functionality yet to be implemented is the inference engine which would manage the basic matching capability in accordance with the hierarchical and recursive structure of process representation.

8.0 CURRENT OUTLOOK

In this section we survey areas needing further development or analysis.

8.1 Representational Requirements

Many processes which would be of interest are not as simply represented as the example of Table 6a.

8.1.1 Parallel Processes/Sequence Overlap

A process necessarily has sequential component processes (the stages). It may well have parallel component processes. For example, in a football game there is a natural sequential structure, i.e., the sequence of plays. But there is also a natural structure of parallel components, namely, the actions of individual players and officials either during a given play or throughout the game. The representation of parallel components adds complexity to the representation of sequential components.

Two formal options are available. The first is to give separate process representations to the parallel components and specify points of coincidence (starting point in common, end point in common, etc. -- there are eight possibilities here). The second is to retain the basic sequential representation but allow some overlap in the occurrence of different stages. If we push the overlap to the limit, we have the same eight possibilities -- otherwise there are seven.

A limited solution can be achieved in many cases by making the overlapping stages into a single stage with two parallel components having the same beginning point (the start of the earliest) and the same ending point (the point where both have ended). This solution was adopted in connection with the many preparatory activities specified in the discursive PS Scenario as successive stages. However, it is clear that a more systematic method of representation is called for.

8.1.2 Optional Stages/Indefinite Number of Stages

The number of stages in a given process may not be fixed (i.e., not the same for all Versions of the process).

(a) For example, the occurrence of a given stage may be contingent on some specified conditions. (E.g., in a convoy with multiple stops, a rest period between stops is optional.)

(b) Or again, the number of stages may be contingent on some specified conditions. (E.g., if the trucks in a convoy deliver their loads one at a time, the number of Stages in the delivery process will vary with the number of trucks in the convoy.)

These cases do call for more complex representation but that does not appear to raise any general difficulties. The issues raised for the inference engine in the computer implementation are potentially more complex.

8.1.3 Fragmentary Representation

The amount of information that must be specified for a complete process presentation ("complete" at the normal level required for processing) is generally large and may be very large. Presumably there will be a significant number of cases where it is not all available, but some of it is, so that we wind up with a fragmentary representation.

8.1.3.1 We have dealt with two <u>systematically</u> incomplete forms of process representation, i.e., Means-Ends description and Task Analysis. However, when we have incomplete knowledge about the process there is no guarantee that we will have all the information for either a Means-Ends description or a Task Analysis. Thus, the availability of these forms of description does not solve the problem in principle.

8.1.3.2 In principle, any degree of representation is better than none, and it may be the case that beyond a certain bare minimum of specification (to get the representation above the noise level, as it were) that will turn out to be the case. As previously, it appears that the most serious problem is likely to be with the automatic processing of such representations rather than with the fragmentary representations as such.

8.1.4 Nameless Relationships

Some of the descriptive schemas, notably those for objects and states of affairs, call for a specification of relationships among components or between a component and the whole of which it is a part. Often, these relationships are unproblematic, but also we have no name for them. For example what is the relationship between

(a) A corporal in a platoon in one regiment and a captain who leads a company in another regiment in the same Division,

(b) The controller in the tower and the navigator of one of the aircraft supervised by the controller,

(c) The turret and the tread of a tank

(d) Etc.

8.1.4.1 In such cases we can introduce purely nominal relationship names, e.g., "the turret-tread relationship," "the controller-navigator relationship," and so on. These specifications may meet the procedural requirements for the representation, but they will be vacuous and uninformative. For that reason standard automatic processing may not be possible.

8.1.4.2 This possibility is very similar to the issue (above) of fragmentary representation. In both cases we are missing the kind of information that is normally required for a complete (i.e., working) representation, and it seems possible that the same resolution will work for both.

8.1.5 Endless Lists

A process may be familiar and non-problematic in a pragmatic sense, but have an infinite set of Options and Versions. For example, any version of "A goes from point B to point C" will have this feature.

In this case our major option is to categorize the Options into a finite set of categories. In order to minimize the arbitrariness of the categories, we may use multiple paradigm cases to anchor these categories.

8.2 Implementation

As noted previously, there are two aspects of the implementation which need further development.

8.2.1 Inference Engine

The simple implementation of inference drawing and of backward and forward chaining is not per se a complicated matter. It is the complexity of the logical structures over which the chaining operates that makes the implementation complex. In the present case, complexities or possible complexities stem from the following.

(a) the hierarchical and recursive structure of process representation (and object and state of affairs representation as well)

(b) the need to process systematically incomplete or simply incomplete representations as well as complete ones

(c) the need to preserve the identity of individuals across representational boundaries

(d) the heterogeneity of Eligibility and Contingency restrictions

(e) the heterogeneity of process representations (simple sequence vs overlap vs parallel; fixed stages vs optional stages; fixed vs variable number of stages).

8.2.2 Simulation/Representation vs Data Base/Knowledge Dictionary

Conceptually, the SA System is designed for simulations. Its most direct use is to reconstruct what is going on on the basis of existing information. However, to operate in this way would require the introduction of certain features which are characteristic of individual persons but which differ significantly among persons. For example, we would include a compatibility or degree of evidence function for drawing conclusions (How much of it do

I have to see before I conclude that <u>this</u> is what is taking place?)

However, a data base for C3I analysis must be accessible to different users, and different users will in general differ in their compatibility criteria (among others). The emphasis, therefore shifts to identifying evidence, degree of compatibility, identifying possibilities, etc. rather than straightforwardly drawing conclusions. In turn, this imposes additional complexities on the functioning system. Ultimately, the problem is that of constructing a Knowledge Base Management System. Fortunately, since the problems are parallel to those of a DBMS, we would expect that many of the solutions will be parallel also.

9.0 RELATION TO OTHER AI APPROACHES

There are about half a dozen currently well known AI approaches. Since they are qualitatively different it is difficult to compare them with one another. It is equally difficult to compare them to the State of Affairs approach. Certain relationships are worth noting.

9.1 Rules

A "rule-based system" is one in which the knowledge base consists primarily or entirely of rules having the form "If A then P," where A may be any set of conditions and P is either a conclusion or an action.

9.1.1 In the system described in Sections 5, 7, and 8 there are two places where some implementation by means of rules is plausible. These are in the Eligibility and Contingency specifications. For example "Option 7.1.10.3 IFF Option 7.1.6.2" or "If Driver (x) then Person (x)".

9.1.2 It is interesting to note that the heterogeneity of content in these specifications was seen as a potential source of difficulty (Section 8.2.1).

9.1.3 This suggests a general principle, namely that a rule-type implementation is most appropriate when the content has no supporting logical structure and must therefore be dealt with in a more or less completely ad hoc way.

Consider, for example, the transformation from "on highway E63" to "on the path from the KMS refinery to Jena air base". The implementation described above involved table lookups and matches. Even when direction of travel and approximate location, e.g. "10 miles west of Gera", are added, as they will need to be, the implementation will be a straightforward table lookup and match. One could write a set of rules for performing these transformations, but that would be pointless and inefficient. It would be pointless because there is enough logical structure in the Path File so that a single set of instructions will work for any of the paths; we therefore do not need a separate instruction (rule) for each case. It would be inefficient because in general, calculation is more efficient than rule implementation.

9.1.3.1 A good analogy is the difference between a mathematical formula and a decision table. If a formula will do the job, that is generally preferable, since the formula itself is informative and easily implemented. It is when there is no formula that we consider the option of taking the possibilities one by one and stipulating what to do in <u>that</u> case.

9.2 Frames

A "Frame" is a notational device for collecting sets of things that "go together". Minsky's example of a birthday party as including children, cakes, candles, and games is a classic one. To introduce a frame with its "slots" filled is equivalent to saying "These things go together -- somehow."

9.2.1 Unlike the case with rules, there is not a portion of the system described above where a frame implementation is especially plausible. On the other hand, there are more places where it would not be particularly implausible.

For example a given Social Practice and the set of Elements for that practice would fit Minsky's birthday party example, i.e. the frame would identify the thing is question and it would contain the things that "go together" as ingredients. Similarly for the list of Object attributes for a given object, the collection of social practices in a given scenario, the set of eligibility requirements for a given Element in a given SP, and the set of mobile objects in the domain.

9.2.2 Note that in all of these possible frame implementations what we are dealing with is something that has a place in a logical structure. (E.g., the Object is one of the objects in the domain and it has certain relationships to other objects and processes; its attributes are not just a collection of attributes -- they are _its_ attributes.)

What we have in each of the examples is a set of things that are represented _as_ going together in certain ways. Because of this, we do not need an additional representation of the fact _that_ they go together _somehow_.

9.2.3 This suggests that frame implementation is most appropriate when we have a collection of things that is not anchored in a structure in which they are related, but _we_ know that they go together and something about how they go together. Under these conditions the grouping must be accomplished in what is, from the standpoint of the rest of the system, a purely arbitrary way and the handling of the group must be done ad hoc.

9.2.4 This result is similar to that for rules. In both cases we have a situation where something has to be done ad hoc in an arbitrary way because there is not enough (structure, content, representational power) in the system to make it anything other than ad hoc.

9.2.5 This review leads to the potentially valuable suggestion that there is a natural complementarity between the representational thrust of the State of Affairs approach and the implementation methodology of rules and frames.

That is that more powerful representation reduces the need for ad hoc procedures, but since man's reach exceeds his grasp, there is likely to be a continuing gap between representational capability and the requirements of particular applications; rules and frames can help bridge that gap.

9.3 Scripts

The resemblance of a script to a social practice representation is obvious. Specifically, a script will (a) mention Elements in the process, (b) identify Stages in the process, and (c) identify some Options for these stages. Moreover, (d) it has a recursive logic insofar as a script may involve sub-scripts.

The differences are fundamental. These include the following.

(a) Scripts are keyed to some customary versions of the process and incorporate a limited range of options; process description is designed to elucidate all possible versions of the process;

(b) Process representations are connected conceptually and notationally to the concepts of object, event, and state of affairs whereas scripts form an isolated conceptual apparatus; as we have seen, these connections are essential even in representing a process as such;

(c) The richness of structure of actual processes requires state of affairs descriptions for intermediate points and for specifying Eligibilities and Contingencies; without comparable resources Script representation of processes makes them oversimplified and mechanical.

Needless to say, what can be represented in a Script can be represented in a social practice description. To be sure, one may not always need as much representation as a social practice description provides.

9.4 Semantic Nets and Inheritance

A semantic net is a schema in which one can introduce some number of discrete items and specify one or more relationships between any pair of items. There is no restriction on what kinds of item or what kinds of relationship can be introduced. Relationships which are commonly included are "isa", i.e., is an instance of, and "hasa", i.e., a possession relationship. Attributes or other pieces of information can be collected at any node.

"Inheritance" arises in connection with "isa". Where there is a genus-species relationship between two items, the attributes of the genus are "inherited" by the species or specimen and therefore do not have to be explicitly associated with it, since the association can be calculated.

9.4.1 "Inheritance" appears to be an implementation technique rather than a representational one. It is not distinctively associated with semantic net methodology; rather, it can be used in almost any representational or computational context.

9.4.2 Because the formal structure of a semantic net is a set of links between pairs of items, it seems clear that the greatest utility of semantic nets is in the representation of two-place relationships. Not surprisingly, it has corresponding limitations.

9.4.2.1 For relationships involving more than two items neither the formal structure nor the notation is suitable. For example, it would be a complicated matter to represent a single entry in the Fact File (Table 7; Appendix E).

9.4.2.2 The representation of a complex logical structure such as is given by the Process schema (BPU) or the Object schema (BOU) or the SA schema (SAU) would be even more complicated. It would also be inefficient.

9.4.2.3 As in the case of rules and frames, where there is a systematic SA system representation, a semantic net is redundant. For example, if we represent several individuals as being related to one another as particular Elements in a given social practice, we do not in addition need a semantic net to tell us <u>that</u> these elements are related and give us the name of the relationship. It is rather when there are no underlying logical structures that we need <u>some</u> way of introducing <u>some</u> structure, and a semantic net will do that.

9.5 GOALS

The GOALS approach is one which structures behaviors in terms of a task analysis. Behaviors are seen as implementing the component tasks in a task analysis, and the latter shows the coherence that exists among the various behaviors.

Task Analysis was discussed above, along with Means-Ends Description, as an incomplete form of Process Description. The two notions of Task Analysis appear to be roughly the same. The general notion appears to be more extensively and systematically developed in the SA System framework. Thus, for example, we systematically distinguish (a) the types of cases where the achievements are sequential and the desired state of affairs is the last achievement in the sequence from (b) the types of cases where the achievements are cumulative and add up to the desired state of affairs (they jointly qualify as the desired state of affairs) from (c) the types of cases where the achievements, including the desired state of affairs, are neither cumulative nor sequential, but are all accomplished simultaneously.

9.6 A review of the relationships to other AI approaches shows two major kinds.

9.6.1 Three of the common approaches (rules, frames, semantic nets) have a relatively loose logical structure which allows heterogeneous elements and relationships to be introduced ad hoc and processed ad hoc. In those areas where a logically structured set of representations such as SA system representations is implemented, such methods are not needed. They might, however, be used to extend the scope of an SA-based system into areas not covered by SA representation.

9.6.2 Two others (scripts, GOALS) appear to correspond quite literally to a portion (process representation) of the SA methodology but in simplified and isolated form. These do not provide any resources which are not present in more highly developed form in the SA system.

9.6.3 The SA model accounts for all the categories of "what there is" and does so at all possible levels of detail and across or within all possible discrete time intervals. It can, in this perspective, be considered as a super-set of knowledge representations in which all other representations can be formulated as restricted versions. This strongly suggests that SA could, in principle, serve as a common schema for knowledge bases through which existing or future expert systems could communicate. Moreover, since SA can be formulated as a normalized relational schema, the practical foundation for implementing SA in this capacity already exists. (N.B. This implies that <u>every</u> existing knowledge representation structure has a normalized relational schema even though it may not yet have been formulated by its advocates.)

10.0 CONCLUSIONS

As noted in Section 8.0, the development of representational requirements and functionality requirements is not considered complete at this time. However, the areas in which potential problems might arise do not appear to offer any decisive difficulties. The fact that a significant proportion of the potential problems are related to the inference engine underlines the importance of developing a fully implemented prototype system to serve as a test bed.

APPENDIX A

INSTALLATIONS AND FACILITIES

A. Bridges
1. Litomerice Hiway River Bridge
2. Zatech RR and Hiway Bridge Comples
3. Zwickauer Mulde River Bridge
4. Durnburg RR and Hiway River Bridge
5. Hohen Hiway Bridge
6. Weimar RR Bridge
7. Gotha RR and Hiway Bridge Complex
8. Kuhndorf Hiway Bridge

B. Power Facilities
1. Stara Sedlo Powr Production Facility
2. Kralupy Power Production Facility
3. Freiburg Nuclear Power Plant
4. Gera Power Production Facility

C. Railroad Facilities
1. Chrasi Rail Turning Yard
2. Nove Sedlo RR Siding and Switching Yard
3. Possneck RR Yard and Break Point
4. Erfurt RR Round House

D. Military Facilities
1. Plzen Barracks
2. Kaster-Tepla Barracks
3. Usti Military Compound
4. Pirna Military Compound
5. Greiz Army Compound
6. Suhl Barracks
7. Eisenach Military Housing Facility

E. Repair and Maintenance Facilities
1. Plzen Tank Park & Maintenance Facility
2. KMS (Karl Marx Stadt) Truck and Cargo Yard
3. Karlovy Vary Mobilizer and Launcher Yard
4. Schneeberg Tank Park
5. Gera Tank Park and Track Repair Facility
6. Manenberg Truck and POL Vehicle Facility
7. Gotha SAM Mobilizer Assembly and Park

F. Munitions Storage Depots
1. Bad Langensalza Munitions Storage Depot

G. Weapons Facilities
1. Prag SAM and PLOT Assembly Facility
2. KMS Munitions Factory

H. Air bases

1. Tchorovice Air base (NE Blovice)
2. Dobris Air base
3. Zwug Air base (Plzen)
4. Prag Air base
5. Panensky Air base (Mrovice)
6. Mukarov Air base (Chomutov)
7. Karlovy Vary Air base
8. Wilkau Air base
9. Jena Air base
10. Blankenhain Air base
11. Waltershausen Air base

I. POL Storage Facilities
1. Bor POL Storage Area
2. Schneeberg POL Storage Facility
3. KMS POL Complex
4. Manenberg POL Storage and Shipment Facility
5. Schleiz Bulk POL Storage Facility
6. Neustadt Jet-A Storage and Pumping Station
7. Grafenrod POL Storage Area
8. Bad Salzungen POL Storage Site

J. Radar and Fire Control Facilities
1. Vodnanv Missile Control Site
2. Rudolstadt Missile Launch Site
3. Lonzig Filter Center (ABM Radar Site)
4. Tambach-Dietharz Fire Control Complex
5. Altenburg Missile Control Complex
6. Bhubel Air Defense Complex
7. Kimze Early Warning Site
8. Vacov GCI Facility
9. Radosice FC Radar Site
10. Mysliv MC Radar Complex
11. Nalozovice FC Radar Complex
12. Horzovsky TYN FC Radar Site
13. Unehle Missile Radar Control Site
14. Marianske Lazne FC Radar Site
15. Dobris GCI Control Facility
16. Bukov FC Radar Site
17. Sokolov FC Radar Site
18. Strov Radar Site
19. Chomutov Missile Control Radar Site
20. Mikulasovice FC Radar Site
21. Vladimerice FC Radar Site
22. Doubov Early Warning Facility
23. Plaven FC Radar Site
24. Beierfeld FC Radar Site
25. Eibenstock Early Warning Facility

26. Usti Missile Control Facility
27. Reichen FC Radar Site
28. Tripris FC Radar Site
29. Schlottwitz FC Radar Site
30. Dippoldiswald Missile Control Site
31. Eppendorf Early Warning Site
32. Nossen FC Radar Site
33. Lobstadt FC Radar Site
34. Korenfahlis FC Radar Site
35. Magdala FC Radar Site
36. Bad Blankenburg FC Radar Site
37. Suhl FC Radar Site
38. Bad Liebenstein FC Radar Site
39. Brientenbach GCI Facility
40. Eissenach FC Radar Facility
41. Weimer GCI Site

K. Missile Sites

1. Stutzerb Missile Site
2. Nalozovice SAM Site
3. Horovsky TYN SAM Site
4. Unehle SAM Site
5. Marianske Lazne SAM Complex
6. Bukov SAM Site
7. Sokolov SAM Site
8. Strov SAM Site
9. Chomutov SAM Site
10. Mikulasovice SAM Site
11. Vladimerice SAM Site
12. Vodnanv SAM Site
13. Radosice SAM Site
14. Plaven SAM Site
15. Beierfeld SAM Site
16. Usti SAM Site
17. Reichen SAM Site
18. Tripris SAM Site
19. Schlottwitz SAM Site
20. Dippoldiswald SAM Site
21. Nossen SAM Site
22. Altenburg SAM Site
23. Lobstadt SAM Site
24. Kohrenfahlis SAM Site
25. Madgala SAM Site
26. Bad Blankenburg SAM Site
27. Suhl SAM Site
28. Bad Liebenstein SAM Site
29. Eissenach SAM Site

APPENDIX B

FORWARD UNITS

FORWARD UNITS

1. 7th Motorized Rifle Division: Plzen Barracks
    a. 88th Artillery Brigade
    b. 27th Motorized Infantry Regiment
    c. 22nd Motorized Infantry Regiment
    d. 26th Tank Brigade

2. 3rd Army: Suhl Barracks
    a. 47th Armored Division
    b. 99th Mechanized Infantry Division
    c. 77th Armored Division

3. 12th Army: Greiz Army Compound
    a. 17th Armored Division
    b. 83rd Mechanized Infantry Brigade
    c. 44th Artillery Brigade
    d. 123rd Mechanized Infantry Brigade

APPENDIX C

PATH FILE

PATH FILE

#	From	To	Seq	Hiway	Dir	Hrs.
1	Manenberg POL Facility	Zwug Air Base	1	174	SE	
	Manenberg POL Facility	Zwug Air Base	2	7	SE	
	Manenberg POL Facility	Zwug Air Base	3	27	S	3.25
2	KMS POL Facility	Wilkau Air Base	1	169	S	1.00
3	KMS POL Facility	Jena Air Base	1	E63	W	
	KMS POL Facility	Jena Air Base	2	88	N	2.50
4	KMS POL Facility	Blankernhain Air Base	1	E63	W	
	KMS POL Facility	Blankernhain Air Base	2	87	SW	
	KMS POL Facility	Blankernhain Air Base	3	85	SE	3.75
5	KMS POL Facility	Waltershn Air Base	1	E63	W	5.00
6	Manenberg POL Facility	Tchorovice Air Base	1	174	E	
	Manenberg POL Facility	Tchorovice Air Base	2	7	E	
	Manenberg POL Facility	Tchorovice Air Base	3	27	S	
	Manenberg POL Facility	Tchorovice Air Base	4	20	S	
	Manenberg POL Facility	Tchorovice Air Base	5	19	E	4.25
7	Manenberg POL Facility	Mukarov Air Base	1	174	E	
	Manenberg POL Facility	Mukarov Air Base	2	7	E	.90
8	Manenberg POL Facility	Prag Air Base	1	174	E	
	Manenberg POL Facility	Prag Air Base	2	7	E	3.10
9	Manenberg POL Facility	Panensky Air Base	1	174	E	
	Manenberg POL Facility	Panensky Air Base	2	7	E	
	Manenberg POL Facility	Panensky Air Base	3	27	S	

#	From	To	Seq	Hiway	Dir	Hrs.
	Manenberg POL Facility	Panensky Air Base	4	20	S	
	Manenberg POL Facility	Panensky Air Base	5	19	E	5.00
10	Manenberg POL Facility	Karlovy Vary Air Base	1	174	E	
	Manenberg POL Facility	Karlovy Vary Air Base	2	7	E	
	Manenberg POL Facility	Karlovy Vary Air Base	3	13	SW	
	Manenberg POL Facility	Karlovy Vary Air Base	4	20	E	2.50
11	KMS POL Facility	Schneeberg POL Facility	1	169	S	1.00
12	KMS POL Facility	Manenberg POL Facility	1	174	SE	.75
13	Greiz Army Compound	Gera Tank Yard	1	92	N	1.00
14	Suhl Barracks	Gera Tank Yard	1	247	N	
	Suhl Barracks	Gera Tank Yard	2	E63	E	4.00
15	Plzen Barracks	Plzen Tank Yard				1.00
16	KMS Munitions Factory	Greiz Army Compound	1	173	S	
	KMS Munitions Factory	Greiz Army Compound	2	92	W	1.75
17	KMS Munitions Factory	Greiz Army Compound	1	E62	S	
	KMS Munitions Factory	Greiz Army Compound	2	92	W	2.00
18	KMS Munitions Factory	Suhl Barracks	1	E63	W	
	KMS Munitions Factory	Suhl Barracks	2	247	S	5.00
19	KMS Munitions Factory	Plzen Barracks	1	174	E	
	KMS Munitions Factory	Plzen Barracks	2	7	E	
	KMS Munitions Factory	Plzen Barracks	3	27	S	5.50
20	Prag Munitions Factory	Greiz Army Compound	1	7	NW	

#	From	To	Seq	Hiway	Dir	Hrs.
	Prag Munitions Factory	Greiz Army Compound	2	174	NW	
	Prag Munitions Factory	Greiz Army Compound	3	173	S	
	Prag Munitions Factory	Greiz Army Compound	4	92	W	6.50
21	Prag Munitions Factory	Suhl Barracks	1	7	NW	
	Prag Munitions Factory	Suhl Barracks	2	174	NW	
	Prag Munitions Factory	Suhl Barracks	3	E63	W	
	Prag Munitions Factory	Suhl Barracks	4	247	S	9.50
22	Prag Munitions Factory	Plzen Barracks	1	E12	SW	2.50
23	Manenberg POL Facility	Dobris Air Base	1	174	E	
	Manenberg POL Facility	Dobris Air Base	2	7	E	
	Manenberg POL Facility	Dobris Air Base	3	12	E	
	Manenberg POL Facility	Dobris Air Base	4	4	S	4.60
24	KMS POL Facility	Bor POL Facility	1	E62	SW	
	KMS POL Facility	Bor POL Facility	2	92	SE	
	KMS POL Facility	Bor POL Facility	3	21	SE	
	KMS POL Facility	Bor POL Facility	4	E12	SW	4.50
25	KMS POL Facility	Bor POL Facility	1	174	E	
	KMS POL Facility	Bor POL Facility	2	7	E	
	KMS POL Facility	Bor POL Facility	3	27	S	
	KMS POL Facility	Bor POL Facility	4	E12	W	5.00
26	KMS POL Facility	Schleiz POL Facility	1	E63	E	
	KMS POL Facility	Schleiz POL Facility	2	2	S	2.60
27	KMS POL Facility	Schleiz POL Facility	1	E62	S	

#	From	To	Seq	Hiway	Dir	Hrs.
	KMS POL Facility	Schleiz POL Facility	2	282	E	
	KMS POL Facility	Schleiz POL Facility	3	2	N	2.50
28	KMS POL Facility	Bad Salzungen POL Facility	1	E63	E	
	KMS POL Facility	Bad Salzungen POL Facility	2	19	S	
	KMS POL Facility	Bad Salzungen POL Facility	3	62	E	4.75
29	KMS POL Facility	Neustadt POL Facility	1	E63	E	
	KMS POL Facility	Neustadt POL Facility	2	2	S	
	KMS POL Facility	Neustadt POL Facility	3	281	E	2.60
30	KMS POL Facility	Neustadt POL Facility	1	173	SW	
	KMS POL Facility	Neustadt POL Facility	2	175	W	
	KMS POL Facility	Neustadt POL Facility	3	2	S	
	KMS POL Facility	Neustadt POL Facility	4	281	E	2.50

APPENDIX D

PREEMPTIVE STRIKE SCENARIO

1: Announce Large Pact Exercise involving ground and air forces

1.1: Option 1: Live Fire Exercise

1.1.1: Stage 1: Use Diplomatic Channels to announce exercise

1.1.2: Stage 2: Call Major Commanders from Germany & Poland

1.1.3: Stage 3: Issue Misleading Exercise oriented messages

1.2: Option 2: Mere Exercise

1.2.1: Stage 1: Call commanders for briefing on "exercise"

1.2.2: Stage 2: Announce exercise locally

1.2.3: Stage 3: Misleading troop movements locally - rehearsal

1.3: Option 3: Command and Control Exercise

1.3.1: Stage 1: Brief Commanders

1.3.2: Stage 2: Conceal Troop Movements by spreading over time

2: Build up resource reserves for operations

2.1: Increase production at Prag munitions plant

2.1.1: Stage 1: Increase rate of arrival of raw materials

2.1.1.1: Option 1: Conceal by holding constant # of shipments but increase load

2.1.1.2: Option 2: Increase both # of shipments and loads

2.1.2: Stage 2: Increase the employment and shifts at plants

2.1.2.1: Option 1: Conceal through pursuing normal resupply announcements

2.1.2.2: Option 2: Conceal through plant expansion announcements

2.1.2.3: Option 3: Recruit and expand operations without explanation

2.1.3: Stage 3: Increase the rate of shipments of munitions

2.1.3.1: Option 1: Resupply at night

2.1.3.2: Option 2: Conceal loads

2.1.3.3: Option 3: Obvious resupply

2.1.4.: Stage 4: Increase the stockpiles of munitions at storage areas

2.1.4.1: Option 1: Conceal through revetment and cover

2.1.4.2: Option 2: Open resupply and buildup

2.2: Fill POL storage tanks at Bor POL storage area

2.2.1: Stage 1: Fill tank trucks at refinery

2.2.1.1: Option 1: Increase tank truck fleet

2.2.1.2: Option 2: Use existing tank truck fleet

2.2.2: Stage 2: Transport fuel to POL storage area

2.2.2.1: Option 1: Independent delivery schedule

2.2.2.2: Option 2: Convoy

2.2.3: Stage 3: Fill storage tanks

2.2.3.1: Option 1: Top off existing tanks

2.2.3.2: Option 2: Supplement existing tanks with bladders & on-site tankers

2.3: Off-load SAM missiles at Nove Sedlo RR siding

2.3.1: Stage 1: Assemble SAM transports at RR siding

2.3.1.1: Option 1: Conceal assembly with sheds

2.3.1.2: Option 2: Assembly in the open

2.3.2: Stage 2: Load SAMs onto transports

2.3.2.1: Option 1: Load SAMs at night

2.3.2.2: Option 2: Load SAMs in open

2.3.3: Stage 3: Move trucks to SAM sites

2.3.3.1: Option 1: Deliver to sites independently as loaded

2.3.3.2: Option 2: Assemble convoys for delivery

2.4: Increase repairs on tanks in Tank Park & Maint Facility

2.4.1: Stage 1: Increase personnel working on tanks in yard

2.4.1.1: Option 1: Increase # of workers in yard

2.4.1.2: Option 2: Increase hours of existing work force

2.4.2: Stage 2: Diagnosis of tanks in yard

2.4.2.1: Option 1: Trained workers examine each of tanks in yard

2.4.2.2: Option 2: Tank crews perform triage

2.4.3: Stage 3: Movement of tanks needing little repair to repair bays

2.4.3.1: Option 1: Tank tow of inoperable vehicles to end of line

2.4.3.2: Option 2: Movement of operable vehicles to front of line

2.4.4: Stage 4: Increase incoming materials needed for tank repair

2.4.4.1: Option 1: Conceal delivery of materials and storage

2.4.4.2: Option 2: Delivery and storage in open

2.4.5: Stage 5: Immediate shipment of tanks following repair

2.4.5.1: Option 1: Pick up by assign tank crews

2.4.5.2: Option 2: Delivery by maintenance staff

2.5: Increase stockpiles of weapons in Bad Langensalza Munitions stores

2.5.1: Stage 1: Increase shipments of incoming munitions to storage

2.5.1.1: Option 1: Conceal the shipment of incoming munitions

2.5.1.2: Option 2: Delivery in open

2.5.2: Stage 2: Construct temporary storage facilities

2.5.2.1: Option 1: Construct as needed

2.5.2.2: Option 2: Total construction of all needed storage

2.5.3: Stage 3: Revet weapons stockpiles

2.5.3.1: Option 1: In conjunction with building

2.5.3.2: Option 2: Independent of building storage facilities

2.6: Increase training at Pirna Military Compound

2.6.1: Stage 1: Increase the number of incoming troops

2.6.1.1: Option 1: Conceal troop movements

2.6.1.2: Option 2: Troop movements in open

2.6.2: Stage 2: Construct temporary housing for incoming troops

2.6.2.1: Option 1: Construct as needed

2.6.2.2: Option 2: Construct in anticipation

2.6.3: Stage 3: Increase the level of training activity in compound

2.6.3.1: Option 1: Normal training activity

2.6.3.2: Option 2: Special training activity

2.7: Fill POL storage tanks at Schneeberg POL Storage Facility

2.7.1: Stage 1: Assemble tank trucks at refinery

2.7.1.1: Option 1: Increase fleet of tank trucks

2.7.1.2: Option 2: Use existing fleet of tank trucks

2.7.2: Stage 2: Fill tank trucks

2.7.2.1: Option 1: Conceal filling operations

2.7.2.2: Option 2: Fill tankers in open

2.7.3: Stage 3: Transport fuel to Schneeberg

2.7.3.1: Option 1: Independent delivery

2.7.3.2: Option 2: Travel in convoy

2.7.4: Stage 4: Fill storage tanks

2.7.4.1: Option 1: Top off existing tanks

2.7.4.2: Option 2: Supplement existing tanks with bladders, etc.

2.8: Increase refining production at Karl Marx Stadt POL Complex

2.8.1: Stage 1: Increase level of incoming shipments of crude oil

2.8.1.1: Option 1: Conceal incoming shipments of crude oil

2.8.1.2: Option 2; Increase openly number of crude arrivals

2.8.2: Stage 2: Increase personnel and shift work at refinery

2.8.2.1: Option 1: Conceal the increase through longer shifts

2.8.2.2: Option 2: Openly recruit workers and add shifts

2.8.3: Stage 3: Increase shipments of POL

2.8.3.1: Option 1: Conceal loading and POL movement

2.8.3.2: Option 2: Openly increase POL shipments

2.9: Increase crude shipments to KMS POL Complex through Manenberg

2.9.1: Stage 1: Increase the incoming rail shipments of crude oil

2.9.1.1: Option 1: Conceal by using larger trains and cars

2.9.1.2: Option 2: Openly increase by more trains

2.9.2: Stage 2: Increase the level of crude off-loading from cars

2.9.2.1: Option 1: Conceal off-loading operations

2.9.2.2: Option 2: Off-load in open

2.10: Increase BIV area for training - Griez Army Compound

2.10.1: Stage 1: Build temporary tent platforms in BIV areas

2.10.1.1: Option 1: Conceal by building as needed

2.10.1.2: Option 2: Build up to meet projected needs

2.10.2: Stage 2: Erect temporary shelters on platforms

2.10.2.1: Option 1: Conceal by erecting as needed

2.10.2.2: Option 2: Erect to meet projected needs

2.11: Fill bladders in Schleiz Bulk POL storage

2.11.1: Stage 1: Fill tanker trucks at refinery

2.11.1.1: Option 1: Increase tank truck fleet

2.11.1.2: Option 2: Conceal by using existing tank truck fleet

2.11.2: Stage 2: Transport fuel to Schleiz

2.11.2.1: Option 1: Independent delivery

2.11.2.2: Option 2: Assemble convoy

2.11.3: Stage 3: Fill bladders from trucks

2.11.3.1: Option 1: Top off current bladders

2.11.3.2: Option 2: Increase the number of storage bladders

2.12: Fill tanks at Grafenrod POL Storage

2.12.1: Stage 1: Fill tank trucks at refinery

2.12.1.1: Option 1: Increase tank truck fleet

2.12.1.2: Option 2: Use current tank truck fleet

2.12.2: Stage 2: Transport fuel to Grafenrod

2.12.2.1: Option 1: Independent delivery

2.12.2.2: Option 2: Assemble and deliver in convoy

2.12.3: Stage 3: Fill storage tanks

2.12.3.1: Option 1: Top off existing tanks

2.12.3.2: Option 2: Augment existing tanks with bladders and tankers

2.13: Reactivate Bad Salzungen POL Storage and fill tanks

2.13.1: Stage 1: Assemble maintenance crews at Bad Salzungen

2.13.1.1: Option 1: Small crew - reconstruction over time

2.13.1.2: Option 2: Larger crew - complete quickly

2.13.2: Stage 2: Test storage tanks

2.13.2.1: Option 1: Test covertly

2.13.2.2: Option 2: Test openly

2.13.3: Stage 3: Fill tank trucks at refinery

2.13.3.1: Option 1: Increase tanker fleet

2.13.3.2: Option 2: Use existing tanker fleet

2.13.4: Stage 4: Transport fuel to Bad Salzungen

2.13.4.1: Option 1: Deliver independently

2.13.4.2: Option 2: Assemble and deliver in convoy

2.13.5: Stage 5: Fill storage tanks

2.13.5.1: Option 1: Fill existing storage tanks

2.13.5.2: Option 2: Augment existing tanks with bladders and tankers

2.13.6: Stage 6: Occupy distribution center

2.13.6.1: Option 1: Conceal occupation and operations

2.13.6.2: Option 2: Occupy and operate openly

2.14: Increase repairs in Tank park and truck repair facility at

(0520T0002)

2.14.1: Stage 1: Increase personnel working on vehicles in yards

2.14.1.1: Option 1: Conceal by lengthening shift

2.14.1.2: Option 2: Add personnel to maintenance crew

2.14.2: Stage 2: Sorting of vehicles awaiting repairs

2.14.2.1: Option 1: Vehicles sorted by maintenance crews

2.14.2.2: Option 2: Vehicles sorted by operation crews

2.14.3: Stage 3: Increased movement of vehicles awaiting repair

2.14.3.1: Option 1: Randomize vehicles sorted by problems

2.14.3.2: Option 2: Line up vehicles for repair

2.14.4: Stage 4: Increase parts shipments

2.14.4.1: Option 1: Conceal parts shipments

2.14.4.2: Option 2: Openly increase parts shipments

2.14.5: Stage 5: Rapid shipment of completed vehicles

2.14.5.1: Option 1: Transport completed vehicles as they are finished

2.14.5.2: Option 2: Assemble completed vehicles into convoys

2.15: Increase output from Pobneck Steel Plant

2.15.1: Stage 1: Increase the shipments of raw materials to steel plant

2.15.1.1: Option 1: Conceal incoming shipments of raw materials

2.15.1.2: Option 2: Increase incoming shipments openly

2.15.2: Stage 2: Increase the personnel and shift work at plant

2.15.2.1: Option 1: Conceal increase by lengthening hours

2.15.2.2: Option 2: Openly recruit and increase personnel

2.15.3: Stage 3: Increase the shipments of finished material

2.15.3.1: Option 1: Ship as material becomes available

2.15.3.2: Option 2: Assemble convoy and ship in bulk

2.16: Increase mobilizer assembly rate

2.16.1: Stage 1: Increase incoming shipments of SAM missiles

2.16.1.1: Option 1: Conceal incoming shipments

2.16.1.2: Option 2: Increase incoming shipments openly

2.16.2: Stage 2: Increase level of missile assembly

2.16.2.1: Option 1: Increase assembly under cover

2.16.2.2: Option 2: Increase assembly openly

2.17: Resupply Air Bases

2.17.1: Stage 1: Increase transport traffic to munitions storage areas

2.17.1.1: Option 1: Increase transport fleet

2.17.1.2: Option 2: Using current transport fleet

2.17.2: Stage 2: Load munitions onto transports

2.17.2.1: Option 1: Under concealment

2.17.2.2: Option 2: Openly

2.17.3: Stage 3: Transport munitions to air bases

2.17.3.1: Option 1: Deliver independently

2.17.3.2: Option 2: Assemble in convoy

2.17.4: Stage 4: Off-load munitions into air base storage units

2.17.4.1: Option 1: Conceal off-loading operations

2.17.4.2: Option 2: Off-load openly

3. Increase readiness status

3.1: Move SA-8 Mobilizers to ends of Litomerice Hiway River Bridge

3.1.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.1.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.1.3: Stage 3: Crews proceed on major highways to installation sites

3.1.4: Stage 4: Crews set up and man SA-8

3.2: Move SA-8 Mobilizers to south end of Zwickau Mulde River Bridge

3.2.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.2.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.2.3: Stage 3: Crews proceed on major highways to installation sites

3.2.4: Stage 4: Crews set up and man SA-8

3.3: Harden distribution building at Stara Sedlo Hydroelectric Plant

3.3.1: Stage 1: Set up construction site at plant

3.3.2: Stage 2: Deliver reinforcing steel to plant

3.3.3: Stage 3: Set reinforcing steel on existing structure

3.3.4: Stage 4: Deliver concrete to plant

3.3.5: Stage 5: Pour concrete on reinforced structure

3.4: Move SA-8 Mobilizers to Bor POL storage area

3.4.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.4.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.4.3: Stage 3: Crews proceed on major highways to installation sites

3.4.4: Stage 4: Crews set up and man SA-8

3.5: Move SA-8 Mobilizers to Kralupy Power Production Facility

3.5.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.5.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.5.3: Stage 3: Crews proceed on major highways to installation sites

3.5.4: Stage 4: Crews set up and man SA-8

3.6: Move SA-8 Mobilizers to Weimar RR Bridge

3.6.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.6.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.6.3: Stage 3: Crews proceed on major highways to installation sites

3.6.4: Stage 4: Crews set up and man SA-8

3.7: Move SA-8 Mobilizers to Kuhndorf Hiway Bridge

3.7.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.7.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.7.3: Stage 3: Crews proceed on major highways to installation sites

3.7.4: Stage 4: Crews set up and man SA-8

3.8: Move SA-8 Mobilizers to Pobneck RR yard and Break point

3.8.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.8.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.8.3: Stage 3: Crews proceed on major highways to installation sites

3.8.4: Stage 4: Crews set up and man SA-8

3.9: Move SA-8 Mobilizers to Gera Power Production Facility

3.9.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.9.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.9.3: Stage 3: Crews proceed on major highways to installation sites

3.9.4: Stage 4: Crews set up and man SA-8

3.10: Revet above ground pipelines from Grafenrod

3.10.1: Stage 1: Move earthmoving equipment to pipeline

3.10.2: Stage 2: Revet the pipeline

3.11: Move SA-8 Mobilizers to Bad Salzungen POL Storage Area

3.11.1: Stage 1: Mobilizer crews to SA-8 assembly sites

3.11.2: Stage 2: Mobilizer crews man SA-8 mobilizers

3.11.3: Stage 3: Crews proceed on major highways to installation sites

3.11.4: Stage 4: Crews set up and man SA-8

3.12: Harden radar facility at Kiemze early warning site

3.12.1: Stage 1: Set up construction site at plant

3.12.2: Stage 2: Deliver reinforcing steel to plant

3.12.3: Stage 3: Set reinforcing steel on existing structure

3.12.4: Stage 4: Deliver concrete to plant

3.12.5: Stage 5: Pour concrete on reinforced structure

3.13: Activate Radnice FC Radar site

3.13.1: Stage 1: Man the Radnice Radar site

3.13.2: Stage 2: Mount radar at site

3.13.3: Stage 3: Prepare pads for air defense system

3.13.4: Stage 4: Install air defense system

3.14: Upgrade Zwug Air Base to Full Status

3.14.1: Stage 1: Set up air field lighting

3.14.2: Stage 2: Set up air servicing control

3.14.3: Stage 3: Construct munitions storage facility

3.14.4: Stage 4: Improve drainage on east end

3.14.5: Stage 5: Increase POL storage

4: Deploy men & equipment

4.1: Distribute munitions from Prag munitions plant

4.1.1: Stage 1: Trucks move to loading dock at north end of building

4.1.2: Stage 2: Load completed SAMs onto trucks

4.1.3: Stage 3: Move trucks with SAMs to missile mobilizer yards

4.2: Vacate Hq Bldg and Barracks at Pleen Barracks

4.2.1: Stage 1: Increase troop trucks into Barracks area

4.2.2: Stage 2: Load troops onto trucks by unit

4.2.3: Stage 3: Transport troops to deployment area

4.2.4: Stage 4: Off-load troops

4.2.5: Stage 5: Return trucks to barracks area

4.3: Deploy HQs 22nd Armored Division Reserve Unit along E12 - target Schwabach

4.3.1: Stage 1: Increase the troop strength of the reserve unit

4.3.2: Stage 2: Assemble the equipment needed to serve the unit

4.3.3: Stage 3: Man the armored division's equipment

4.3.4: Stage 4: Deploy the equipment and troops by unit

4.3.5: Stage 5: Move the Hqs to the field

4.4: Deploy HQs 4th Motorized Rifle Division along #26 - target Regensburg

4.4.1: Stage 1: Assemble Division's motorized equipment

4.4.2: Stage 2: Load troops onto equipment by unit

4.4.3: Stage 3: Deploy equipment by unit

4.5: Truck spare SAM missiles to deployment sites

4.5.1: Stage 1: Assemble trucks at rail facility

4.5.2: Stage 2: Load spare SAMs onto trucks

4.5.3: Stage 3: Move trucks from rail facility to a SAM site

4.6: Deploy vehicles from Tank park & Maint facility to units

4.6.1: Stage 1: Move tank crews to Tank park

4.6.2: Stage 2: Man operational vehicles

4.6.3: Stage 3: Move operational vehicles to units

4.6.4: Stage 4: Randomly space vehicles awaiting maintenance

4.6.5: Stage 5: Revet vehicles awaiting maintenance

4.7: Deploy vehicles from truck and cargo yard to units

4.7.1: Stage 1: Assemble drivers for vehicles in truck yard

4.7.2: Stage 2: Move vehicles loaded with munitions to units

4.7.3: Stage 3: Move troop carriers to barracks for troop deployment

4.7.4: Stage 4: Assemble trucks to off-load material from rail yard

4.8: Deploy SA-8 Mobilizers and missiles from mobilizer yard to defense sites

4.8.1: Stage 1: Assemble SAM crews at mobilizer yard

4.8.2: Stage 2: Man mobilizers

4.8.3: Stage 3: Move mobilizers to deployment areas

4.9: Deploy tanks from tank park to units

4.9.1: Stage 1: Move tank crews to tank park

4.9.2: Stage 2: Man armored vehicles in tank park

4.9.3: Stage 3: Move armored vehicles to units for deployment

4.10: Resupply all units in Bad Langengaliza munitions area

4.10.1: Stage 1: Move supply trucks to munitions storage area

4.10.2: Stage 2: Load munitions onto the trucks

4.10.3: Stage 3: Move munitions to deployed units

4.11: Deploy 6th Armored Division along #13 - initial target Wurzburg

4.11.1: Stage 1: Assign troops to units

4.11.2: Stage 2: Assemble troop transports in barracks area

4.11.3: Stage 3: Load troops onto transports

4.11.4: Stage 4: Deploy troops to units

4.12: Deploy 90th Lt. Inf. along #13 following the 6th - initial target Bamberg

4.12.1: Stage 1: Assign troops to units

4.12.2: Stage 2: Assemble troop transports in barracks area

4.12.3: Stage 3: Load troops onto transports

4.12.4: Stage 4: Deploy troops to units

4.13: Deploy Dresden units along E62 at junction 92 - initial target Schweinfurt

4.13.1: Stage 1: Assign troops to units

4.13.2: Stage 2: Assemble troop transports in barracks area

4.13.3: Stage 3: Load troops onto transports

4.13.4: Stage 4: Deploy troops to units

4.14: Deploy Self-propelled units along 89 - initial target Fulda

4.14.1: Stage 1: Assign troops to units

4.14.2: Stage 2: Assemble troop transports in barracks area

4.14.3: Stage 3: Load troops onto transports

4.14.4: Stage 4: Deploy troops to units

4.15: Deploy airborne troops to air field for deployment to junction E70 & A2

4.15.1: Stage 1: Distribute munitions to self-propelled units

4.15.2: Stage 2: Assemble self-propelled units and man vehicles

4.15.3: Stage 3: Move self-propelled units to deployment areas

4.15.4: Stage 4: Assemble troop transports for support personnel

4.15.5: Stage 5: Man troop transports

4.15.6: Stage 6: Deploy support troops

4.16: Deploy tanks and trucks from (0520T0002) to units

4.16.1: Stage 1: Man tanks that have been serviced

4.16.2: Stage 2: Deploy operational tanks to units

4.16.3: Stage 3: Randomly distribute tanks still awaiting repair

4.16.4: Stage 4: Increase stocks of steel in yard

5: Explore Allied Readiness

5.1: Paradigm 1: Single aircraft - single exploration

5.1.1: Stage 1.1: M-21 takes off from strip alert at Tchorovice Air Base

5.1.2: Stage 1.2: Climbs to alt 20,000m, heading 270 deg, speed 500 knots

5.1.2.1: Option 1.2.1: Alt < 30,000m, 150 deg< h <290 deg, speed > stall

5.1.3: Stage 1.3: 20km from border - drop below radar threshold

5.1.4: Stage 1.4: 10km from border - above radar threshold heading 90 degrees

5.1.4.1: Option 1.4.1: Alt > 900m, 10 deg< h <170 deg, speed > stall

5.1.5: Stage 1.5: Return to base

5.1.5.1: Option 1.5.1: Proceed to alternate Base

5.1.6: Stage 1.6: Land M-21

5.1.7: Stage 1.7: Refuel M-21

5.1.8: Stage 1.8: Redeploy at strip readiness site

5.1.8.1: Option 1.8.1: Redeploy off strip readiness site

5.2: Paradigm 2: Single aircraft - multiple explorations

5.2.1: Stage 2.1: M-21 takes off from strip alert at Tchorovice Air Base

5.2.2: Stage 2.2: Climbs to alt 20,000m, heading 270 deg, speed 500 knots

5.2.2.1: Option 2.2.1: Alt < 30,000m, 150 deg< h <290 deg, speed > stall

5.2.3: Stage 2.3: 20km from border - drop below radar threshold

5.2.4: Stage 2.4: 10km from border - above radar threshold heading 90 degrees

5.2.4.1: Option 2.4.1: Alt > 900m, 10 deg< h <170 deg, speed > stall

5.2.5: Stage 2.5: Turn to heading 180 degrees, alt 5000m, speed 600 knots

5.2.6: Stage 2.6: Drop below radar threshold

5.2.7: Stage 2.7: Climb to 1000m, heading 270 degrees, speed 800 knots

5.2.8: Stage 2.8: Drop below radar threshold

5.2.9: Stage 2.9: Turn to 90 degrees, speed 600 knots, climb to 10,000m

5.2.10: Stage 2.10: Return to base

5.2.10.1: Option 2.10.1: Proceed to alternate Base

5.2.11: Stage 2.11: Land M-21

5.2.12: Stage 2.12: Refuel M-21

5.2.13: Stage 2.13: Redeploy at strip readiness site

5.2.13.1: Option 2.13.1: Redeploy off strip readiness site

5.3: Paradigm 3: Multiple aircraft - single exploration

5.3.1: Stage 3.1: 5 M-21 aircraft taxi from strip alert at

Tchorovice Air Base

5.3.1.1: Option 3.1.1: 2 < # of aircraft < 10

5.3.2: Stage 3.2: M-21 aircraft take off and assemble at 10,000m

5.3.2.1: Option 3.2.1: Alt > 10,000m

5.3.3: Stage 3.3: Formation heading 270 degrees, speed 500 knots

5.3.4: Stage 3.4: 30km from border, drop to 3000m, speed 600 knots

5.3.5: Stage 3.5: 20km from border, drop to 1000m, speed 700 knots

5.3.6: Stage 3.6: 15km from border, drop below radar threshold

5.3.7: Stage 3.7: Turn to heading 100 degrees, climb to 10,000m

5.3.8: Stage 3.8: Return to Air Base

5.3.8.1: Option 3.8.1: Proceed to alternative Air Base

5.3.9: Stage 3.9: Land in formation

5.3.9.1: Option 3.9.1: Land sequentially

5.3.10: Stage 3.10: Refuel simultaneously

5.3.10.1: Option 3.10.1: Refuel sequentially

5.3.11: Stage 3.11: Redeploy on strip alert

5.3.11.1: Option 3.11.1: Redeploy off strip alert

5.4: Paradigm 4: Multiple aircraft - multiple explorations

5.4.1: Stage 4.1: 5 M-21 aircraft taxi from strip alert at Tchorovice Air Base

5.4.1.1: Option 4.1.1: 2 < # of aircraft < 10

5.4.2: Stage 4.2: M-21 aircraft take off and assemble at 10,000m

5.4.2.1: Option 4.2.1: Alt > 10,000m

5.4.3: Stage 4.3: Formation heading 270 degrees, speed 500 knots

5.4.4: Stage 4.4: 30km from border, drop to 3000m, speed 600 knots

5.4.5: Stage 4.5: Disperse 1km apart to the south in parallel

5.4.6: Stage 4.6: 20km from border, drop to 1000m, speed 700 knots

5.4.7: Stage 4.7: 15km from border, drop below radar threshold

5.4.8: Stage 4.8: Turn to heading 100 degrees, climb to 10,000m

5.4.9: Stage 4.9: Return to Air Base

5.4.9.1: Option 4.9.1: Proceed to alternative Air Base

5.4.10: Stage 4.10: Land in formation

5.4.10.1: Option 4.10.1: Land sequentially

5.4.11: Stage 4.11: Refuel simultaneously

5.4.11.1: Option 4.11.1: Refuel sequentially

5.4.12: Stage 4.12: Redeploy on strip alert

5.4.12.1: Option 4.12.1: Redeploy off strip alert

6: Attack

6.1: Relocate personnel from training area

6.1.1: Stage 1: Assign men to operational units

6.1.1.1: Option 1: Assign men to new units

6.1.1.2: Option 2: Assign men to existing units

6.1.2: Stage 2: Form men into units

6.1.2.1: Option 1: Simultaneously

6.1.2.2: Option 2: By unit

6.1.3: Stage 3: Assemble troop movements

6.1.3.1: Option 1: To move all trainees

6.1.3.2: Option 2: To move part of trainees

6.1.4: Stage 4: Load troops onto transports

6.1.4.1: Option 1: By units

6.1.4.2: Option 2: Simultaneously

6.1.5: Stage 5: Transport troops to operational units

6.1.5.1: Option 1: Independently

6.1.5.2: Option 2: In convoys

6.2: Begin feint in direction of Marianske Lazne

6.2.1: Paradigm 1: With live fire

6.2.1.1: Dispatch of Wing in support of strategic preemptive strike

6.2.1.1.1: Aircraft leave various airfields

6.2.1.1.1.1: Paradigm 1: Coordinated departure

6.2.1.1.1.1.1: Ground Crews prepare aircraft at Tchorovice Airfield

6.2.1.1.1.1.1.1: Option 1: Fueled and loaded with operational ordinance

6.2.1.1.1.1.2: Pilots briefed on mission

6.2.1.1.1.1.2.1: Option 1: With security

6.2.1.1.1.1.3: Pilots man planes

6.2.1.1.1.1.4: Aircraft take off from Tchorovice Airfield at prearranged time

6.2.1.1.1.1.4.1: Option 1: Aircraft depart independently

6.2.1.1.1.1.4.2: Option 2: Aircraft depart in squadrons

6.2.1.1.1.2: Paradigm 2: Independent departure

6.2.1.1.1.2.1: Ground Crews prepare aircraft at Tchorovice

Airfield

6.2.1.1.1.2.1.1: Option 1: Fueled and loaded with operational ordinance

6.2.1.1.1.2.2: Pilots briefed on mission

6.2.1.1.1.2.2.1: Option 1: With security

6.2.1.1.1.2.3: Pilots man planes

6.2.1.1.1.2.4: Aircraft take off from Tchorovice Airfield as they come on line

6.2.1.1.1.2.4.1: Option 1: Aircraft depart independently

6.2.1.1.1.2.4.2: Option 2: Aircraft depart in squadrons

6.2.1.1.2: Aircraft rendezvous at prearranged location

6.2.1.1.2.1: Paradigm 1: Independent flight plans

6.2.1.1.2.1.1: Aircraft proceed on individual flight plans to Plasy

6.2.1.1.2.1.2: Aircraft hold at Plasy

6.2.1.1.2.1.3: Squadrons reform at Plasy

6.2.1.1.2.1.4: Escort aircraft meet with strategic bombers at Plasy

6.2.1.1.2.2: Paradigm 2: Group flight plans

6.2.1.1.2.2.1: Squadrons form at 2000m over Tchorovice airfield

6.2.1.1.2.2.2: Formation turns to 210 degrees

6.2.1.1.2.2.3: Formation holds at Plasy

6.2.1.1.2.2.4: Escort aircraft meet with strategic bombers at Plasy

6.2.1.1.3: Aircraft deploy in escort positions

6.2.1.1.3.1: Paradigm 1: Combat aircraft lead preemptive strike

6.2.1.1.3.1.1: Aircraft turn to 270 degrees for target

6.2.1.1.3.1.2: Combat aircraft proceed at 1000m in advance of bombers

6.2.1.1.3.1.3: Bombers proceed at 2000m to targets

6.2.1.1.3.2: Paradigm 2: Combat aircraft screen strategic bombers

6.2.1.1.3.2.1: Bombers turn to 270 degrees to target at 3000m

6.2.1.1.3.2.2: Combat aircraft deploy by squadrons to flanks of bombers

6.2.1.1.3.2.3: Combat aircraft proceed at 3500m

6.2.1.1.4: Aircraft accompany strategic bombers to target

6.2.1.1.4.1: Paradigm 1: Efficient course

6.2.1.1.4.1.1: Aircraft proceed to target at 270 degrees, 3000m, 300 knots

6.2.1.1.4.1.2: On approach, aircraft drop to 2000m for bombing run

6.2.1.1.4.1.2.1: Option 1: High altitude mission

6.2.1.1.4.1.2.2: Option 2: Low altitude mission

6.2.1.1.4.2: Paradigm 2: Evasive course

6.2.1.1.4.2.1: Aircraft proceed at 200 degrees, 1000m, 300 knots

6.2.1.1.4.2.2: Aircraft alter course to 340 degrees, 1500m, 400 knots

6.2.1.1.4.2.2.1: Option 1: One course change

6.2.1.1.4.2.2.2: Option 2: Multiple course changes

6.2.1.1.4.2.3: Aircraft adopt final 270 degrees, 600m, 500 knots

6.2.1.1.5: Aircraft fly live fire mission

6.2.1.1.5.1: Paradigm 1: Live fire mission

6.2.1.1.5.1.1: Option 1: Combat aircraft fly interdiction

6.2.1.1.5.1.1.1: On approach, combat aircraft take lead

6.2.1.1.5.1.1.2: Combat aircraft engage prearranged targets

6.2.1.1.5.1.1.3: Combat aircraft head to Plana holding site

6.2.1.1.5.1.1.4: Combat aircraft hold at Plana holding site

6.2.1.1.5.1.1.5: Strategic bombers fly bombing run

6.2.1.1.5.1.1.5.1: Option 1.5.1: High altitude 5000m

6.2.1.1.5.1.1.5.2: Option 1.5.2: Low altitude 600m

6.2.1.1.5.1.1.6: Strategic bombers engage targets

6.2.1.1.5.1.1.7: Strategic bombers proceed to Plana holding site

6.2.1.1.5.1.2: Option 2: Combat aircraft fly support

6.2.1.1.5.1.2.1: On approach, combat aircraft screen flanks at 1000m

6.2.1.1.5.1.2.2: Strategic bombers fly bombing run

6.2.1.1.5.1.2.2.1: Option 2.2.1: High altitude 5000m

6.2.1.1.5.1.2.2.2: Option 2.2.2: Low altitude 600m

6.2.1.1.5.1.2.3: Strategic bombers engage targets

6.2.1.1.6: Aircraft regroup after bombing run

6.2.1.1.6.1: Strategic bombers join with combat aircraft at Plana holding site

6.2.1.1.6.2: Combat aircraft deploy on screening position

6.2.1.1.6.3: Formation adopts 90 degrees, 5000m, 400 knots to Plasy

6.2.1.1.7: Aircraft return to initial bases

6.2.1.1.7.1: At Plasy site aircraft adopt 70 degrees to Tchorovice airfield

6.2.1.1.7.2: Aircraft land at Tchorovice airfield

6.2.1.1.7.2.1: Option 1: By squadron

6.2.1.1.7.2.2: Option 2: Independently

6.2.1.1.7.3: Aircraft serviced

6.2.2: Paradigm 2: Abortive feint

6.2.2.1: Lead elements of 22nd Armored reserve move toward site

6.2.2.2: Lead elements of 4th motorized division move toward site

6.2.2.3: Following elements proceed to border

6.3: Troops begin attack from north to south

6.3.1: Paradigm 1: Lead with air strike

6.3.1.1: Dispatch of Wing in support of strategic preemptive strike

6.3.1.1.1: Aircraft leave various airfields

6.3.1.1.1.1: Paradigm 1: Coordinated departure

6.3.1.1.1.1.1: Ground Crews prepare aircraft at [Airfield]

6.3.1.1.1.1.1.1: Option 1: Fueled and loaded with operational ordinance

6.3.1.1.1.1.2: Pilots briefed on mission

6.3.1.1.1.1.2.1: Option 1: With security

6.3.1.1.1.1.3: Pilots man planes

6.3.1.1.1.1.4: Aircraft take off from [airfield] at prearranged time

6.3.1.1.1.1.4.1: Option 1: Aircraft depart independently

6.3.1.1.1.1.4.2: Option 2: Aircraft depart in squadrons

6.3.1.1.1.2: Paradigm 2: Independent departure

6.3.1.1.1.2.1: Ground Crews prepare aircraft at [airfield]

6.3.1.1.1.2.1.1: Option 1: Fueled and loaded with operational ordinance

6.3.1.1.1.2.2: Pilots briefed on mission

6.3.1.1.1.2.2.1: Option 1: With security

6.3.1.1.1.2.3: Pilots man planes

6.3.1.1.1.2.4: Aircraft take off from [airfield] as they come on line

6.3.1.1.1.2.4.1: Option 1: Aircraft depart independently

6.3.1.1.1.2.4.2: Option 2: Aircraft depart in squadrons

6.3.1.1.2: Aircraft rendezvous at prearranged location

6.3.1.1.2.1: Paradigm 1: Independent flight plans

6.3.1.1.2.1.1: Aircraft proceed on individual flight plans to [rendezvous]

6.3.1.1.2.1.2: Aircraft hold at [rendezvous]

6.3.1.1.2.1.3: Squadrons reform at [rendezvous]

6.3.1.1.2.1.4: Escort aircraft meet with strategic bombers at [rendezvous]

6.3.1.1.2.2: Paradigm 2: Group flight plans

6.3.1.1.2.2.1: Squadrons form at [alt] over [airfield]

6.3.1.1.2.2.2: Formation turns to [heading]

6.3.1.1.2.2.3: Formation holds at [rendezvous]

6.3.1.1.2.2.4: Escort aircraft meet with strategic bombers at [rendezvous]

6.3.1.1.3: Aircraft deploy in escort positions

6.3.1.1.3.1: Paradigm 1: Combat aircraft lead preemptive strike

6.3.1.1.3.1.1: Aircraft turn to [heading] for target

6.3.1.1.3.1.2: Combat aircraft proceed at [alt] in advance of bombers

6.3.1.1.3.1.3: Bombers proceed at [alt] to targets

6.3.1.1.3.2: Paradigm 2: Combat aircraft screen strategic bombers

6.3.1.1.3.2.1: Bombers turn to [heading] to target at [alt]

6.3.1.1.3.2.2: Combat aircraft deploy by squadrons to flanks of bombers

6.3.1.1.3.2.3: Combat aircraft proceed at [alt]

6.3.1.1.4: Aircraft accompany strategic bombers to target

6.3.1.1.4.1: Paradigm 1: Efficient course

6.3.1.1.4.1.1: Aircraft proceed to target at [heading], [alt], [speed]

6.3.1.1.4.1.2: On approach, aircraft drop to [alt] for bombing run

6.3.1.1.4.1.2.1: Option 1: High altitude mission

6.3.1.1.4.1.2.2: Option 2: Low altitude mission

6.3.1.1.4.2: Paradigm 2: Evasive course

6.3.1.1.4.2.1: Aircraft proceed at [heading], [alt], [speed]

6.3.1.1.4.2.2: Aircraft alter course to [heading], [alt], [speed]

6.3.1.1.4.2.2.1: Option 1: One course change

6.3.1.1.4.2.2.2: Option 2: Multiple course changes

6.3.1.1.4.2.3: Aircraft adopt final [heading], [alt], [speed]

6.3.1.1.5: Aircraft engage targets

6.3.1.1.5.1: Paradigm 1: Live fire mission

6.3.1.1.5.1.1: Option 1: Combat aircraft fly interdiction

6.3.1.1.5.1.1.1: On approach, combat aircraft take lead

6.3.1.1.5.1.1.2: Combat aircraft engage prearranged targets

6.3.1.1.5.1.1.3: Combat aircraft head to [holding site]

6.3.1.1.5.1.1.4: Combat aircraft hold at [holding site]

6.3.1.1.5.1.1.5: Strategic bombers fly bombing run

6.3.1.1.5.1.1.5.1: Option 1.5.1: High altitude [alt]

6.3.1.1.5.1.1.5.2: Option 1.5.2: Low altitude [alt]

6.3.1.1.5.1.1.6: Strategic bombers engage targets

6.3.1.1.5.1.1.7: Strategic bombers proceed to [holding site]

6.3.1.1.5.1.2: Option 2: Combat aircraft fly support

6.3.1.1.5.1.2.1: On approach, combat aircraft screen flanks at [alt]

6.3.1.1.5.1.2.2: Strategic bombers fly bombing run

6.3.1.1.5.1.2.2.1: Option 2.2.1: High altitude [alt]

6.3.1.1.5.1.2.2.2: Option 2.2.2: Low altitude [alt]

6.3.1.1.5.1.2.3: Strategic bombers engage targets

6.3.1.1.6: Aircraft regroup after bombing run

6.3.1.1.6.1: Strategic bombers join with combat aircraft at [holding site]

6.3.1.1.6.2: Combat aircraft deploy on screening position

6.3.1.1.6.3: Formation adopts [heading], [alt], [speed] to [dispersal site]

6.3.1.1.7: Aircraft return to initial bases

6.3.1.1.7.1: At [dispersal site] aircraft adopt [heading] to [airfield]

6.3.1.1.7.2: Aircraft land at [airfield]

6.3.1.1.7.2.1: Option 1: By squadron

6.3.1.1.7.2.2: Option 2: Independently

6.3.1.1.7.3: Aircraft serviced

6.3.2: Paradigm 2: Lead with ground assault

6.3.2.1: Self-propelled units to Fulda

6.3.2.2: Airborne units to junction E70 & A2

6.3.2.3: Headquarters units to Scheinfurt

6.3.2.4: 6th Armored units to Witzburg

6.3.2.5: Hq 22nd Armored to Schwabach

6.3.2.6: 4th Motorized units to Regenburg

APPENDIX E

FACT FILE (DATA BASE)

Appendix E - Page 1 - Left Side

FACT TYPE	ACTION	ELEMENT	ELEMENT	ELEMENT
1	Depart	FTruck	MS Munitions Factory	*
2	On	FTrucks	Hiway 92	*
3	Arrive	FTrucks	Ammo	Greiz Army Compound
4	Depart	TTrucks	KMS POL Facility	*
7	Arrive	TTrucks	Zwug Air Base	*
6	Arrive	TTrucks	BOR POL Facility	*
4	Depart	TTrucks	Manenberg POL Facility	*
14	Parked at	Tanks	Gera Tank Yard	*
4	Depart	TTrucks	KMS POL Facility	*
4	Depart	TTrucks	Manenberg POL Facility	*
5	On	TTrucks	Hiway 19	*
6	Arrive	TTrucks	Schleiz POL Facility	*
3	Arrive	FTrucks	Ammo	Suhl Barracks
10	Depart	TATrucks	Greiz Army Compound	*
12	Arrive	TATrucks	Gera Tank Yard	*
4	Depart	TTrucks	KMS POL Facility	*
4	Depart	TTrucks	Manenberg POL Facility	*
4	Depart	TTrucks	Manenberg POL Facility	*
4	Depart	TTrucks	KMS POL Facility	*
1	Depart	FTrucks	KMS Munitions Facility	*
10	Depart	TATrucks	Plzen Barracks	*
7	Arrive	TTrucks	Makarov Air Base	*
7	Arrive	TTrucks	Prag Air Base	*
3	Arrive	FTrucks	Plzen Barracks	*
4	Depart	TTrucks	KMS POL Facility	*
7	Arrive	TTrucks	Panensky Air Base	*
4	Depart	TTrucks	Manenberg POL Facility	*

Appendix E - Page 1 - Right Side

Element	Quantity	Units	Date	Time	Cred
*	6	Ea	01-04-88	0905	2
*	6	Ea	01-04-88	1040	2
*	6	Ea	01-04-88	1100	2
*	16	Ea	01-04-88	0800	2
*	11	Ea	01-04-88	1230	2
*	5	Ea	01-04-88	1140	2
*	11	Ea	01-04-88	0900	2
*	15	Ea	01-04-88	1035	2
*	10	Ea	01-05-88	0812	2
*	6	Ea	01-05-88	0700	1
*	6	Ea	01-05-88	1100	2
*	5	Ea	01-05-88	1100	2
*	3	Ea	01-05-88	1215	2
*	5	Ea	01-05-88	0920	1
*	5	Ea	01-05-88	1025	2
*	5	Ea	01-06-88	0820	2
*	5	Ea	01-06-88	0910	2
*	15	Ea	01-06-88	0800	2
*	5	Ea	01-06-88	0712	1
*	3	Ea	01-06-88	0705	2
*	7	Ea	01-06-88	0910	2
*	5	Ea	01-06-88	1015	2
*	15	Ea	01-06-88	1115	2
*	3	Ea	01-06-88	1250	2
*	11	Ea	01-07-88	0715	2
*	9	Ea	01-07-88	1215	2
*	11	Ea	01-07-88	0910	2

Appendix E - Page 2 - Left Side

FACT TYPE	ACTION	ELEMENT	ELEMENT	ELEMENT
7	Arrives	TTruck	Karlovy Vary Air Base	*
6	Arrives	TTrucks	Neustadt POL Facility	*
1	Departs	FTrucks	KMS Munitions Factor	*
10	Departs	TATrucks	Suhl Barracks	*
3	Arrives	FTrucks	Ammo	Plzen Barracks
12	Arrives	TATrucks	Gera Tank Yard	*
5	On	TTrucks	Hiway 2	*
5	On	TTrucks	Hiway E63	*
5	On	TTrucks	Hiway 7	*
7	Arrives	TTrucks	Dobris Air Base	*
6	Arrives	TTrucks	Schneeberg POL Facility	*
5	On	TTrucks	Hiway 169	*
14	Parked at	Tanks	Gera Tank Yard	*
14	Parked at	Tanks	Plzen Tank Yard	*
5	On	TTrucks	Hiway 169	*
7	Arrives	TTrucks	Wilkan Air Base	*
4	Departs	TTrucks	Manenberg POL Facility	*
5	On	TTrucks	Hiway 27	*
6	Arrives	TTrucks	Bor POL Facility	*
3	Arrives	FTrucks	Greiz Army Compound	*
14	Parked at	Tanks	Gera Tank Yard	*
4	Depart	TTrucks	KMS POL Facility	*
4	Depart	TTrucks	Manenberg POL Facility	*
4	Depart	TTrucks	KMS POL Facility	*
1	Depart	FTrucks	KMS Munitions Factory	*
10	Depart	TATrucks	Greiz Army Compound	*
3	Arrives	FTrucks	Ammo	Suhl Barracks

Appendix E - Page 2 - Right Side

Element	Quantity	Units	Date	Time	Cred
*	11	Ea	01-07-88	1150	2
*	11	Ea	01-07-88	1040	2
*	3	Ea	01-07-88	0700	2
*	6	Ea	01-07-88	0820	2
*	3	Ea	01-07-88	1245	2
*	6	Ea	01-07-88	1215	2
*	11	Ea	01-07-88	1000	2
*	11	Ea	01-07-88	1130	2
*	8	Ea	01-08-88	0804	2
*	8	Ea	01-08-88	1202	2
*	10	Ea	01-08-88	1020	2
*	10	Ea	01-08-88	0913	2
*	13	Ea	01-08-88	0820	2
*	11	Ea	01-08-88	1515	2
*	15	Ea	01-11-88	0845	2
*	15	Ea	01-11-88	0910	2
*	10	Ea	01-11-88	0910	2
*	10	Ea	01-11-88	1130	2
*	6	Ea	01-11-88	1144	2
*	5	Ea	01-11-88	1100	2
*	14	Ea	01-11-88	1500	2
*	10	Ea	01-12-88	0810	2
*	7	Ea	01-12-88	0710	2
*	5	Ea	01-12-88	0820	2
*	3	Ea	01-12-88	0715	2
*	6	Ea	01-12-88	0900	2
*	3	Ea	01-12-88	1230	2

Appendix E - Page 3 - Left Side

FACT TYPE	ACTION	ELEMENT	ELEMENT	ELEMENT
7	Arrives	TTruck	Jena Air Base	*
6	Arrives	TTrucks	Schleiz POL Facility	*
14	Parked at	Tanks	Plzen Tank Yard	*
4	Departs	TTrucks	KMS POL Facility	*
4	Departs	TFTrucks	Manenberg POL Facility	*
1	Departs	FTrucks	KMS Munitions Factory	*
10	Departs	TATrucks	Plzen Barracks	*
5	On	TTrucks	Hiway 7	*
5	On	TTrucks	E63	*
3	Arrives	FTrucks	Ammo	Plzen Barracks
14	Parked at	Tanks	Gera Tank Yard	*
4	Departs	TTrucks	KMS POL Facility	*
4	Departs	TTrucks	Manenberg POL Facility	*
4	Departs	TTrucks	Manenberg POL Facility	*
10	Departs	TATrucks	Suhl Barracks	*
3	Arrives	FTrucks	Ammo	Plzen Barracks
7	Arrives	TTrucks	Waltershn Air Base	*
6	Arrives	TTrucks	Neustadt POL Facility	*
5	On	TTrucks	Hiway 2	*
5	On	TTrucks	Hiway 19	*
7	Arrives	TTrucks	Karlovy Vary Air Base	*
7	Arrives	TTrucks	Dobris Air Base	*
6	Arrives	TTrucks	Schneeberg POL Facility	*
5	On	TTrucks	Hiway 12	*
4	Depart	TTrucks	KMS POL Factory	*
14	Parked at	Tanks	Gera Tank Yard	*
14	Parked at	Tanks	Plzen Tank Yard	*

Appendix E - Page 3 - Right Side

Element	Quantity	Units	Date	Time	Cred
*	10	Ea	01-12-88	1045	2
*	5	Ea	01-12-88	1100	2
*	12	Ea	01-12-88	1430	2
*	5	Ea	01-13-88	0820	2
*	5	Ea	01-13-88	0910	2
*	3	Ea	01-13-88	0700	2
*	9	Ea	01-13-88	0915	2
*	17	Ea	01-13-88	1100	1
*	5	Ea	01-13-88	1035	2
*	3	Ea	01-13-88	1250	2
*	13	Ea	01-13-88	1420	2
*	14	Ea	01-14-88	0715	2
*	10	Ea	01-14-88	0700	2
*	14	Ea	01-14-88	0920	2
*	7	Ea	01-14-88	0820	2
*	3	Ea	01-14-88	1300	2
*	14	Ea	01-14-88	1230	2
*	12	Ea	01-14-88	1105	2
*	14	Ea	01-14-88	1000	2
*	10	Ea	01-14-88	1200	2
*	14	Ea	01-14-88	1200	2
*	8	Ea	01-15-88	1150	2
*	13	Ea	01-15-88	1015	2
*	8	Ea	01-15-88	1045	2
*	13	Ea	01-15-88	0905	2
*	12	Ea	01-15-88	1420	2
*	13	Ea	01-15-88	1605	2

Appendix E - Page 4 - Left Side

FACT TYPE	ACTION	ELEMENT	ELEMENT	ELEMENT
7	Arrives	TTrucks	Wilkin Air Base	*
7	Arrives	TTrucks	Zwug Air Base	*
4	Departs	TTrucks	KMS POL Facility	*
6	Arrives	TTrucks	Bor POL Facility	*
3	Arrives	FTrucks	Greiz Army Compound	Ammo
14	Parked at	Tanks	Gera Tank Yard	*
4	Departs	TTrucks	Manenberg POL Facility	*
7	Arrives	TTrucks	Jena Air Base	*
7	Arrives	TTrucks	Tchorovice Air Base	*
6	Arrives	TTrucks	Schleiz POL Facility	*
4	Departs	TTrucks	KMS POL Facility	*
5	On	TTrucks	Hiway 2	*
2	On	FTrucks	Hiway E63	*
1	Departs	FTrucks	KMS Munitions Factory	*
10	Departs	TATrucks	Greiz Army Compound	*
12	Arrives	TATrucks	Gera Tank Yard	*
4	Departs	TTrucks	KMS POL Facility	*
4	Departs	TTrucks	Manenberg POL Facility	*
4	Departs	TTrucks	Manenberg POL Facility	*
6	Arrives	TTrucks	Bad Salzungen POL Facility	*
3	Arrives	FTrucks	Ammo	Plzen Barracks
12	Arrives	TATrucks	Plzen Tank Yard	*
5	On	TTrucks	Hiway 174	*
5	On	TTrucks	Hiway 2	*
5	On	TTrucks	Hiway 2	*
7	Arrive	TTrucks	Waltershn Air Base	*
7	Arrives	TTrucks	Panensky Air Base	*

Appendix E - Page 4 - Right Side

Element	Quantity	Units	Date	Time	Cred
*	18	Ea	01-18-88	0918	2
*	13	Ea	01-18-88	1230	2
*	8	Ea	01-18-88	0705	2
*	8	Ea	01-18-88	1140	2
*	5	Ea	01-18-88	1100	2
*	15	Ea	01-18-88	1620	2
*	13	Ea	01-19-88	0905	2
*	12	Ea	01-19-88	1040	1
*	9	Ea	01-19-88	1120	2
*	9	Ea	01-19-88	1050	2
*	12	Ea	01-19-88	0800	2
*	12	Ea	01-19-88	1003	2
*	4	Ea	01-19-88	1120	2
*	4	Ea	01-19-88	0710	2
*	8	Ea	01-19-88	0900	2
*	8	Ea	01-19-88	1015	2
*	8	Ea	01-20-88	0810	2
*	9	Ea	01-20-88	0905	2
*	18	Ea	01-20-88	0800	2
*	7	Ea	01-20-88	1155	2
*	3	Ea	01-20-88	1233	2
*	10	Ea	01-20-88	1018	2
*	18	Ea	01-20-88	0850	2
*	9	Ea	01-20-88	0950	2
*	9	Ea	01-20-88	1130	2
*	13	Ea	01-21-88	1220	2
*	10	Ea	01-21-88	1215	2

Appendix E - Page 5 - Left Side

FACT TYPE	ACTION	ELEMENT	ELEMENT	ELEMENT
17	Deliver	TTrucks	Karlovy Vary Air Base	POL
4	Departs	TTrucks	Manenberg POL Facility	*
4	Departs	TTrucks	KMS POL Facility	*
16	Deliver	TTrucks	POL	Meistadt POL Facility
1	Departs	TATrucks	Suhl Barracks	Tanks
5	On	TTrucks	Hiway E63	*
5	On	TTrucks	Hiway 174	*
5	On	TTrucks	Hiway 19	*
14	Parked at	Tanks	Plzen Tank Yard	*
17	Deliver	TTrucks	POL	Dobris Air Base
4	Departs	TTrucks	Manenberg POL Facility	*
4	Departs	TTrucks	KMS POL Facility	*
5	On	TTrucks	Hiway 169	*
5	On	TTrucks	Hiway 12	*
14	Parked at	Tanks	Gera Tank Yard	*

Appendix E - Page 5 - Right Side

Element	Quantity	Units	Date	Time	Cred
*	15000	Gal	01-21-88	*	2
*	10	Ea	01-21-88	0900	2
*	13	Ea	01-21-88	0715	2
*	20000	Gal	01-21-88	*	2
*	6	Ea	01-21-88	0815	2
*	13	Ea	01-21-88	1110	2
*	10	Ea	01-21-88	0740	2
*	13	Ea	01-21-88	0930	2
*	12	Ea	01-21-88	1420	2
*	12000	Gal	01-22-88	*	3
*	8	Ea	01-22-88	0715	2
*	14	Ea	01-22-88	0910	2
*	14	Ea	01-22-88	0945	2
*	8	Ea	01-22-88	1100	2
*	11	Ea	01-22-88	1540	2

APPENDIX F

SOFTWARE USER MANUAL

TABLE OF CONTENTS

1. INTRODUCTION 172

2. FUNCTIONAL OVERVIEW 172

3. USER INTERFACE ENVIRONMENT 173

4. FILES 175

5. WINDOWS 177

6. TRANSACTIONS 178

7. RELATIONS 181

8. QUERIES 182

9. SCAN 186

10. JSPACE 188

11. ENGINES 189

12. MISCELLANEOUS COMMANDS 199

13. MACRO COMMANDS 206

14. INSTALLATION 207

15. REFERENCES 209

LIST OF ILLUSTRATIONS

FIGURE 1 - INITIAL SCREEN 194

FIGURE 2 - SCAN MENU 194

FIGURE 3 - QUERY MENU 195

FIGURE 4 - JSPACE MENU 195

FIGURE 5 - TRANSACTION MENU 196

FIGURE 6 - WINDOW MENU 196

FIGURE 7 - FILE MENU 197

FIGURE 8 - RELATION MENU 197

FIGURE 9 - ENGINE MENU 198

1. INTRODUCTION

This appendix describes a stand-alone software system that was developed, in part, to support the contractual effort. Although no software was specifically required as a deliverable, its development and use was indicated early in the effort. The system is entitled "TT: Tables and Text" and supports both relations (tables) and discursive data (text) in a single, unified environment. As such, it could properly be considered a hypertext [1] system or, alternatively, a database system that supports discursive, textual domains.

2. FUNCTIONAL OVERVIEW

The knowledge base for a State of Affairs system must eventually be stored and maintained in a form compatible with existing database systems for processing by various inference engines. The process of developing this knowledge base typically begins with narrative descriptions of the scenario; proceeds to more structured textual descriptions (e.g., outlines, pseudo-code); and concludes with an admixture of structured tables (for use by the inference engines) and related discursive explications (for the convenience of the user). Such a progression requires the support of a word processor, a text editor, and a database system. And because the discursive information persists even in the final knowledge base, all three are required concurrently and continually throughout the effort. While it may be possible to maintain the knowledge base in three separate systems with appropriate interfaces, our experience has been that a single system possessing the combined functionality of all three is clearly called for.

A system which we refer to as a Knowledge Dictionary System was implemented for the purpose of creating, updating, maintaining and searching a State of Affairs knowledge base. This system combines, in a single integrated environment, functions of word processing, text editing, and database management. The functionality of this system is as follows.

3. USER INTERFACE ENVIRONMENT

Upon entering the system the user is presented with the screen illustrated in Figure 1. The first line is referred to throughout as either the Status Line or Command Line. Underneath the Command Line is the Menu Bar, and beneath that is Window 1 which is opened to a file called TEMP1. The Status Line conveys information about the currently active window. In the illustration, this indicates that Window 1 is opened to a file called TEMP1 which is 127 characters wide and contains only 1 row. In Window 1 the cursor is positioned at the beginning of that row which is obviously blank. The Menu Bar is an array of headings, each of which correspond to an object upon which the system can operate. These objects are Scans, Queries, JSpaces, Transactions, Windows, Files, Relations and Engines. The functions of which the system is capable can be accessed in any of three ways.

3.1. Menu Bar. The menu bar represents the major functions that can be executed by the user. The menu bar is activated by F10 and the submenus are selected either by positioning with the arrow keys and pressing CR or by typing the capitalized letter of the selection. ESC backs out of any menu selection and a second ESC deactivates the menu bar. For example, typing F10, moving the cursor across the File heading, pressing CR, moving the cursor down to the Quit selection and pressing CR again will yield a prompt on the Command line "^KX Exit (Y/N)?" Typing a "y" will exit the system. The same function can be accessed by typing F10, F (for File), Q (for quit). Not all of the available functions are accessible from the menu system. The sub-menus are illustrated in Figures 2 through 9.

3.2. Control Keys. Every available function of the system is accessible via special keys (e.g., arrow keys, tab, etc.) and one or two key sequences of control characters. In the above example, note that the prompt was preceded by "^KX" which is the control key sequence for the Quit function. I.e., holding down the Ctrl key, type kx and the same prompt will appear.

3.3. Macro Commands. Any combination of commands and responses to prompts can be invoked from the keyboard. A Macro Command is an ASCII file of a sequence of keystrokes and referred to by the name of the file in which the keystrokes are stored. A Macro Command is executed by typing F9 (which returns the prompt "Execute File:") followed by the file name and CR. This will cause the system to respond sequentially to each character in the file as if it had been typed from the keyboard.

4. FILES

A file is a data stream which can be viewed or modified through a window. Files are of three types: Tables, Text (a table with one column), and Documents (a "table" with no columns). To both the system and the user all types are pretty much the same except some functions behave differently for each type and not all functions are applicable to all types. The operations that can be performed with files are as follows.

4.1. Open File. A file can be OPENed in a window. This makes the file visible to the user and available for searching or updating.

4.2. Close File. A file can be CLOSEd, i.e., made invisible and unavailable for searching or updating.

4.3. Read File. Internally, files are maintained in a unique structure to facilitate the various operations of the system. But if the user has a standard ASCII file (e.g., produced by another program) it can be read in its entirety into the system and converted.

4.4. Write File. A user can also WRITE an entire file in standard ASCII format.

4.5. Top File. This positions the cursor to the first character of the first row of the file.

4.6. Bottom File. This positions the cursor to the last character of the last row of the file.

4.7. Synchronize File. Unlike typical word processors or text editors, the system does not read the entire file into RAM for processing. It, instead, employs a caching scheme whereby only the data most needed is resident in memory. Unless the user takes advantage of Transaction Management (see below) it is advisable to periodically SYNCHronize the file; i.e., force all changes to be recorded on the disk; just as one would periodically save a file when using a text editor.

4.8. Quit. This is a quick shutdown function to synchronize and close all opened files and exit to the operating system.

5. WINDOWS

Windows are viewing areas on the screen consisting of 1 or more display lines. New or empty windows are attached to temporary files called TEMP1, TEMP2, etc. and will remain so until the window or file is closed (if the user wants to save a temporary file, the system will prompt him for a permanent name). The operations that can be performed with windows are as follows.

5.1. Select Window. To operate on data in a window it must be SELECTed, i.e., made the current window. The cursor appears in only the current window and the Status Line always refers to the current window.

5.2. Open Window. A new window can be OPENed, i.e., made to appear on the screen. The user must always specify how many rows the window will contain (its length) and which existing window it will overlay. A window can be opened directly to a file or it can be opened empty in which case the user must specify how wide (in characters) the window should be.

5.3. Close Window. An existing window can be CLOSEd (removed from the screen). If the window is opened to a user file, the file will be closed automatically. If the window is opened to a non-empty temporary file, the user will be prompted for a file name if he wants that file saved.

5.4. Clear Window. Clearing a window is identical to closing it except that the window will remain on the screen attached to an empty temporary file.

5.5. Link Window. A window can be linked to the same file to which an existing window is already opened for the purpose of having two or more areas of the file visible at the same time. Changes made to the file from any of the windows will appear in all of the other windows (immediately, if the viewing area overlaps).

6. TRANSACTIONS

Changes to the data may, at the users option, be governed by a transaction management protocol. A transaction is a sequence of changes that are: atomic; i.e., either they all occur or none of them do and durable; i.e., once made, they can only be changed by another transaction.

6.1. Begin Transaction. BEGIN starts a transaction. It has the effect of establishing an opening parentheses in an equation in that everything within the parentheses will be treated as an atomic unit. When a transaction is begun, the status line will be augmented to indicate the number of rows that have been changed since the transaction began (Log) and the remaining capacity of the log, in rows, to absorb further changes (once a row has been changed, further changes to it do not require additional log capacity).

6.2. Commit Transaction. COMMIT ends a transaction favorably; i.e., it has the effect of establishing the closing parentheses and then performing all the changes that occurred within the parentheses by applying the changes to the disk.

6.3. Abort Transaction. ABORT ends a transaction unfavorably; i.e., it has the effect of establishing the closing parentheses and then undoing all the changes that occurred within the parentheses by restoring the disk to the way it was before the transaction began.

6.4. Enter Transaction. If a transaction has begun in one window, the user may desire to have changes in another window be a part of that transaction (i.e., they will have the same destiny as that transaction when it ends). This is accomplished by entering the transaction.

6.5. Leave Transaction. Once a transaction has been entered, the user may desire that the fate of the changes in a window not be the same as that of the transaction. This is accomplished by leaving the transaction.

6.6. Share Transaction. If two or more windows are open to the same file (or in a multi-user environment, two or more users have a window open to the same file) the user who begins the transaction can designate that the transaction be shared. I.e., the file will appear in all windows as if the transaction were going to be committed.

6.7. Exclusive Transaction. By contrast to a shared transaction, this option provides that other windows open to the file will see it as it was before the transaction began no rows in the file that have already been accessed by the transaction can be changed until the transaction ends.

6.8. Scan Transaction. This function allows the creation of a scan (see below) of all rows that have been changed since the transaction began. (If a row has been deleted, it will not be included in the scan even though it will be un-deleted if the transaction is aborted.)

7. RELATIONS

Relations (tables) are the common format in which all files are viewed. A relation consists of zero or more columns, each having a unique name within the file (Documents, i.e., tables with zero columns, do not have column names and cannot be the object of functions that require column names).

7.1. Create Relation. New relations can be CREATEd. To do so, the user responds to the prompt "Header:" by typing dashes (-) followed by the column name followed by more dashes and a vertical bar (¦) as the column separator. The dashes are optional as in the following example:

```
----Name----¦-------Address---------¦AREA¦-ZIP-¦
```

7.2. Align Column(s). The data in the columns can be re-ALIGNed - either left, right or centered.

7.3. Sort by Column(s). The rows can be SORTed according to the value of one or more columns.

7.4. Add Column(s). A new column can be ADDed TO a relation. The system will prompt for the name of the column after which the new column(s) is to be added and then prompt for the header which is specified in the same way as for creating a relation.

7.5. Drop Column(s). An existing column can be DROPped FROM a relation.

7.6. Change Column. An existing column can be CHANGEd. The system will prompt for a new header as in adding a column. The new header may be smaller or larger in width than the old one.

7.7. Switch Columns. Two existing columns may be positionally interchanged.

7.8. Unique. This command removes adjacent rows from a relation that have identical values for the specified columns. In response to the "Unique:" prompt the user types column1, column2,...columnN. Specifying "s" in response to the "Options:" prompt will cause the table to be sorted on those columns before the command is executed.

8. QUERIES

Queries (searches) may be executed by a user upon a file. The results of a query may be represented by: positioning the tables in the windows (the default); creating a scan; or writing to another window. All query types involve matching of expressions to values and the following wild-card syntax is supported: "\?" matches any single character; "\#" matches any single digit; "*" matches any string of characters; "\$" matches any string of digits; and "\\" matches "\". Furthermore, all query types support the following "Options:"

"a" - locate all occurrences (the default is to locate just the first or next occurrence)

"o" - output the results (the system will prompt for the window number)

"s" - include the results in a scan

"x" - execute the search immediately (the default is to wait for an explicit command to begin)

"i" - ignore case (the default is case sensitive) "w" - whole word matches only, i.e., the matching value must be delimited at both ends (e.g., by space, comma, period, etc.).

Once a query has been defined it may be re-executed using either the NEXT entry on the menu or the Ctrl-L key [2].

8.1. Find. FIND locates occurrences of a specified value anywhere in a file. In response to the prompt "Find:" the user types the pattern followed by CR.

8.2. Replace. REPLACE locates occurrences of a specified value anywhere in a file and replaces it with a specified value. In response to the prompt "Find:" the user types the pattern followed by CR. Then, in response to the prompt "Replace with:" the user types the value to be substituted and CR. There is an additional option unique to the Replace command. The option "n" indicates to perform the replacement without confirmation by the user. The default is to ask the user for each occurrence whether to perform the replacement or not.

8.3. Keyword. KEYWORD locates the conjunction of specified values in the same row anywhere in the file. In response to the prompt "Keyword:" the user types value1 & value2 &...& valueN.

8.4. Select. SELECT locates rows whose column values match a specified value. In response to the prompt "Select:" the user types column1 = value1 & column2 = value2 &...& columnN = valueN.

8.5. Project. PROJECT takes all the rows from a table but only the specified columns and writes them to another window. The "a" and "o" options are both default and mandatory. In response to the prompt "Project:" the user types column1, column2,...,columnN.

8.6. Join. JOIN concatenates two tables based on matching values in the specified columns of each. In response to the "Join:" prompt the user types A.column1 = B.column1 & A.column2 = B.column2 & ... where A denotes the number of the window to be joined to B which is the current window.

8.7. Union. UNION combines two tables of the identical format into a single table. The "a" and "o" options are default and cannot be overridden. In response to the prompt "Union:" the user types the window number containing the tables to be unionized with the table in the current window [3].

8.8. Difference. DIFFERENCE searches two tables of identical format for rows that are based on not matching values in the specified columns of each (i.e., a row in either table qualifies if there is no row in the other table that has equal values in the specified columns). In response to the "Difference:" prompt the user types A.column1 # B.column1 & A.column2 # B.column2 & ... where A denotes the number of the window to be compared to B which is the current window [4].

8.9. Next. NEXT re-executes the currently defined query.

8.10. Clear. CLEAR clears the definition of the currently defined query.

9. SCAN

A scan is a subset of a table. It can be produced by a query, an engine, or the user. A scan is also persistent; i.e., the scan is not lost when a file is closed. Only one scan per table is presently supported. When a scan exists, those rows that are included in the scan are displayed in **highlighted** text on the screen. Scans can be created in any type of file including Documents. In this capacity, it similar to the "block" commands found in word processors but is slightly more flexible in that while a "block" must be contiguous, a scan can include any lines scattered throughout the file.

9.1. Include. A user can manually INCLUDE the current row (the row the cursor is on) in a scan.

9.2. Exclude. A user can manually EXCLUDE the current row from a scan.

9.3. Begin/End. A user can manually include a contiguous set of rows in a scan by Beginning a scan at the current row, moving the cursor to some subsequent row, and ENDing the scan. This will include all the rows, inclusively, between the begin and end commands in the scan. Once a BEGIN command is issued, an "S" will appear on the Status Line until the END command is issued.

9.4. Clear. A user can CLEAR a scan; i.e., exclude all the rows from the scan.

9.5. Read. A user can READ a scan from another window into the current window. If the scan to be read is from a table, the current window must either be a table of identical format or empty.

9.6. Write. A user can WRITE a scan in the current window to another window. If the current window is a table, the destination window must be a table of identical format or empty.

9.7. **Next/Prior.** A user can position the cursor at the NEXT or PRIOR row of a scan.

9.8. **Delete.** A user can DELETE a scan in its entirety from the table in the current window.

9.9. **Move.** A user can MOVE a scan in the current window to another place in the file. All the rows in the scan, whether contiguous or not, will be deleted from their present position and inserted contiguously at the new position somewhere else in the table.

9.10. **Copy.** A user can COPY a scan in the current window to another place in the file. All the rows in the scan, whether contiguous or not, will be inserted contiguously at the specified position in the file.

10. JSPACE

Judgement Spaces are "inferential domains" through which queries may, at the user's option, be resolved; i.e., the "matching" of expressions is not done by lexical analysis, but by locating values that are close together in a multi--dimensional factor space based on user judgments. A JSpace can be CREATED from a table whose columns represent a set of user-selected variables and whose rows correspond to the rating of rows (or a sample of rows) from an existing table against each variable. It can also be REMOVEd (destroyed). If a user OPENs an JSpace, subsequent queries will use that JSpace until it's CLOSEd. Changes to data in the tables are not reflected in the indexes until they are synchronized. The currently open index can be SYNCHronized or the user can SYNCHronize ALL the indexes that pertain to the data, whether opened or not [5].

11. ENGINES

Inference engines are, in effect, very complex queries; complex principally in the fact that they cannot be quantified. I.e., unlike relational queries for which the criteria can be specified in advance, the criterion is dynamic and self-modifying as the query proceeds [6]. The system has only one such engine presently implemented, the CLOSURE engine, but several more are anticipated as a result of the Phase I investigation and will be briefly described as if they existed [7].

11.1. Closure. The Closure engine takes one row of single table and produces the closure [8] of that table based upon two or more columns and constant values. While Closure will accept any table as input, it is only sensible when the table denotes a part-whole relationship such as a work breakdown in which one column represents the Task Name and the other represents the Subtask Name and the Subtask also appears somewhere else in the table as another Task with its own Subtasks, ad infinitum. In this case, Closure can either list all of the Subtasks (and their Subtasks, etc.) required for a Task, or for a given Subtask, list all the Tasks (and Tasks of which those are Subtasks, etc.) of which it could be a Subtask. In response to the prompt "Closure:" the user types A.column1 = A.column2 & A.column3 = A.column4 & ... & A.columnK = A.columnJ where A denotes either the number of the current window or, optionally on the right side of the equal sign, a backslash "\". The option "o" is default and cannot be overridden. The result is obtained by writing the current row to the destination window and then, iteratively: (a) joining the current row of the destination window with the current window based on the column matches; (b) appending the rows in the current window that qualify to the table in the destination window; and (c) joining the next row in the destination window as in (a). This process continues until there is no next row in the destination window. Expressions of the type A.columnK = \.valueK are interpreted to mean that, in addition to the column matches specified, column K must have value K. The table in the destination window is appended with a column with the header of --^--¦ which indicates the level in the part-whole hierarchy at which result occurs.

11.2. Match. The Match engine produces a join between a specified table and the lowest (or highest) level subset of a Closure for which the join is non-empty. Using the Task/Subtask relation and a table that relates Tasks and Task Managers, Match produces a table of all the lowest (or highest) level Managers that must approve a change to a specific Subtask.

11.3. Part-Whole. The Part-Whole engine performs a Closure and Match for every entry in a part-whole relation at a specified level (or set of levels or all levels). Given the Task/Subtask relation and a table relating some subset of the Tasks and their Status (e.g., completed, in-work, etc.) the Part-Whole engine produces a table of every Task in the project and its status (the status of some Tasks may not be inferable based on limited information).

11.4. Deviance. The Deviance engine performs a difference between a specified table and either a Match or a Part-Whole. Given the Task/Subtask relation and the Status table above and a table relating each Task with its scheduled Status, the Deviance engine produces the list of Tasks that are at variance with the schedule.

11.5. Temporal. The Temporal engine takes a specified table (or subset of it) and produces a table that represents a "wait-for" graph. While it can be applied to any table, it is only sensible if the table denotes a precedence relationship such as a table relating a Task and its Status to another Task and its Status in which the first [Task, Status] is only possible when the second [Task, Status] is true (Status can be any combination of columns including numerical data). Given such a table, the Temporal engine will produce a table listing each Task, the Tasks it is waiting for, the Tasks those are waiting for, etc.

11.6. Resource. The Resource engine takes a specified table (or subset of it) and produces a table that represents a queue. While it can be applied to any table, it is only sensible if the table denotes a consumption relationship such as a table relating a Task and its Status to a quantity of a resource in which the [Task, Status] is only possible when the quantity of the resource is available (Status can be any combination of columns including numerical data). Given such a table, the Resource engine will produce a table listing each Resource, the Tasks waiting for that resource, the Tasks waiting for those Tasks because they need that resource too, etc.

11.7. **Begin.** Begin prompts the user to select an engine, specify the parameters for that engine, and starts the engine processing.

11.8. **End.** End stops an engine and clears its parameters.

11.9. **Checkpoint.** Checkpoints suspends an engine and causes it to write a record of all parameters it requires to resume processing from where it left off.

11.10. **Restart.** Restart prompts the user to select an engine which was checkpointed, and causes the engine to read its restart record and resume processing.

FIGURE 1 - INITIAL SCREEN

```
|1|TEMP1,127,1          IX:     A I S  ** TX Log:0    Cap:1394 **
 Scan  Query  JSpace  Transact  Window  File  Relation  Engine
|1|----------------------------TEMP1----------------------------
```

FIGURE 2 - SCAN MENU

```
|1|TEMP1,127,1          IX:     A I S  ** TX Log:0    Cap:1394 **
 Scan  Query  JSpace  Transact  Window  File  Relation  Engine
 Include      |-------------------TEMP1-------------------------
 eXclude
 Begin
 End
 cLear
 Read
 Next
 Prior
 Delete
 Move
 Copy
 Write
```

FIGURE 3 - QUERY MENU

```
|1|TEMP1,127,1          IX:    A I S  ** TX Log:0    Cap:1394 **
+-------------------------------------------------------------------+
| Scan  Query  JSpace  Transact  Window  File  Relation  Engine     |
+-------------------------------------------------------------------+
|1|------| Find          |-----------TEMP1-------------------------
         | Replace       |
         | Keyword       |
         | Select        |
         | Project       |
         | Join          |
         | Next          |
         | Clear         |
         | Union         |
         | Difference    |
         +---------------+
```

FIGURE 4 - JSPACE MENU

```
|1|TEMP1,127,1          IX:    A I S  ** TX Log:0    Cap:1394 **
+-------------------------------------------------------------------+
| Scan  Query  JSpace  Transact  Window  File  Relation  Engine     |
+-------------------------------------------------------------------+
|1|-------------| Create        |---TEMP1--------------------------
                | Remove        |
                | Open          |
                | Close         |
                | Synch         |
                | synch All     |
                +---------------+
```

FIGURE 5 – TRANSACTION MENU

```
|1|TEMP1,127,1          IX:     A I S  ** TX Log:0     Cap:1394 **
 Scan  Query  JSpace  Transact  Window  File  Relation  Engine
|1|                   Begin
                      Commit
                      Abort
                      Enter
                      Leave
                      Share
                      eXclusive
                      Scan
```

FIGURE 6 – WINDOW MENU

```
|1|TEMP1,127,1          IX:     A I S  ** TX Log:0     Cap:1394 **
 Scan  Query  JSpace  Transact  Window  File  Relation  Engine
|1|                             Select
                                Open
                                Close
                                cleaR
                                Link
```

FIGURE 7 - FILE MENU

```
|1|TEMP1,127,1          IX:     A I S  ** TX Log:0    Cap:1394 **
┌─────────────────────────────────────────────────────────────────┐
│ Scan  Query  JSpace  Transact  Window  File  Relation  Engine   │
└─────────────────────────────────────────────────────────────────┘
|1|───────────────────────────TEMP1───│ Open          │──────────
                                      │ Close         │  |
                                      │ Read          │
                                      │ Write         │
                                      │ Top           │
                                      │ Bottom        │
                                      │ Synch         │
                                      │ Quit          │
                                      └───────────────┘
```

FIGURE 8 - RELATION MENU

```
|1|TEMP1,127,1          IX:     A I S  ** TX Log:0    Cap:1394 **
┌─────────────────────────────────────────────────────────────────┐
│ Scan  Query  JSpace  Transact  Window  File  Relation  Engine   │
└─────────────────────────────────────────────────────────────────┘
|1|───────────────────────────TEMP1──────────│ Create        │────
                                             │ aLign         │ |
                                             │ Sort          │
                                             │ Add to        │
                                             │ Drop from     │
                                             │ cHange        │
                                             │ sWitch        │
                                             │ Unique        │
                                             └───────────────┘
```

FIGURE 9 - ENGINE MENU

```
|1|TEMP1,127,1          IX:     A I S  ** TX Log:0    Cap:1394 **
+---------------------------------------------------------------------+
| Scan  Query  JSpace  Transact  Window  File  Relation  Engine       |
+---------------------------------------------------------+-----------+
|1|-----------------------TEMP1---------------------------| Closure   |
                                                          | Match     |
                                                          | Part-Whole|
                                                          | Deviance  |
                                                          | Temporal  |
                                                          | Resource  |
                                                          | Begin     |
                                                          | End       |
                                                          | checKpoint|
                                                          | reStart   |
                                                          +-----------+
```

12. MISCELLANEOUS COMMANDS

All of the commands that can be executed via the pulldown menus may also be executed by a one or two key control sequence. There are many additional commands that can only be executed by a control key sequence. The following is the complete command set currently implemented.

12.1. **One Key Commands.** These are the commands accessed by holding the CTRL key down while typing a single character or pressing one of the specially designated keys (e.g., TAB) on the keyboard.

^B or Shift-Tab: Tab to the previous column of a table or a previous position in a document. If the system is in Auto-Tab mode (an "A" appears on the Status Line), the cursor will position under the first non-blank character in the previous row.

^C or PgDn: Page Down; i.e., move down the file one window length.

^D or Rt Arrow: Cursor Right; i.e., move the cursor right one character.

^E or Up Arrow: Cursor Up; i.e., move the cursor up the file one row.

^G or Del: Delete Right Character; i.e., delete the character under the cursor and shift all the remaining characters after the cursor left one position.

^H or BS: Delete Left Character; i.e., delete the character to the left of the cursor and shift all the remaining characters after the cursor left one position.

^I or Tab: Tab to the next column of a table or the next position in a document. If the system is in Auto-Tab mode (an "A" appears on the Status Line), the cursor will position under the first non-blank character in the previous row.

^J or Home/End: Beginning/End of Row.

^K: Prefix to the set of two key commands that begin with ^K. It is not necessary to hold the CTRL key down for the second character.

^L: Repeat Last Query; i.e., execute the currently defined query in the current window again. See the endnote on query processing for a more complete description of how the ^L command operates.

^M or CR: New Row; i.e., insert a new blank row between the row the cursor is on and the following row. Characters to the right of the cursor, if any, will be deleted from the current row and moved to the new row automatically.

^N: Insert Row; i.e., similar to CR except that no characters will be deleted from the current row and the new row will be empty.

^O: Prefix to the set of two key commands that begin with ^O. It is not necessary to hold the CTRL key down for the second character.

^P: Insert Control Character; i.e., in order to place a control character in the data (e.g., a printer command), type Ctrl-P and the next character will be interpreted as a control character. This is <u>not</u> necessary for macros. The macro command processor interprets any upper case letter as a control character and anything else as itself.

^Q: Prefix to the set of two key commands that begin with ^Q. It is not necessary to hold the CTRL key down for the second character.

^R or PgUp: Window Up; i.e., move backward in the file one window length.

^S or Lft Arrow: Cursor Left; i.e., move the cursor to the left one character position.

^U: Abort Any Command in Progress. While typing on the Command Line, the system will act as if the command had never been started. If the command (e.g., a search) is already in progress, the system will stop processing at the first occasion in which everything is properly synchronized.

^V or Ins: Toggle Inset/Typeover Mode; i.e., in Typeover Mode the character typed will replace the character under the cursor; in Insert mode (an "I" appears on the Status Line) the character typed will be inserted under the cursor and all characters to the right of the cursor will be shifted right one character position.

^W: Scroll Window Up; i.e., move the window backward one row in the file.

^X or Dn Arrow: Cursor Down; i.e., move the cursor to the next row in the file.

^Y: Delete Row; i.e., remove the row the cursor is on from the file.

^Z: Scroll Window Down; i.e., move the window forward one row in the file.

12.2. ^K Prefix Commands. These are the commands accessed _after_ typing Ctrl-K. It is not necessary to hold the CTRL key down when typing the second letter of the command. It will always be interpreted as a control character.

^KB: Begin Scan (previously described).

^KC: Copy Scan (previously described).

^KE: Exclude Row from Scan (previously described).

^KG: Read Scan (previously described).

^KH: Clear Scan (previously described).

^KI: Include Row in Scan (previously described).

^KK: End Scan (previously described).

^KL: Prior Row in Scan (previously described).

^KN: Next Row in Scan (previously described).

^KO: Open File in Window (previously described).

^KP: Write Scan to Window (previously described).

^KR: Read ASCII Text File (previously described).

^KT: Define Tab Width; i.e., set the number of character positions to tab in a document when not in Auto-Tab mode.

^KU: Abort Command (same as ^U).

^KV: Move Scan (previously described).

^KW: Write ASCII Text File (previously described).

^KX: Quit and Exit System (previously described).

^KY: Delete Scan (previously described).

^KZ: Close File (previously described).

12.3. **^O Prefix Commands.** These are the commands accessed _after_ typing Ctrl-O. It is not necessary to hold the CTRL key down when typing the second letter of the command. It will always be interpreted as a control character.

^OA: Abort Transaction (previously described).

^OB: Begin Transaction (previously described).

^OC: Commit Transaction (previously described).

^OD: Leave Transaction (previously described).

^OE: Enter Transaction (previously described).

^OG: Select Window (previously described).

^OI: Tab to Next Column (previously described).

^OJ: Link Window (previously described).

^OK: Change Case; i.e., if the character under the cursor is lower case it will be changed to upper case and conversely.

^OL: Center Text; i.e., center the text under the cursor in the column.

^ON: Include Transaction in Scan (previously described).

^OO: Open Window (previously described).

^OS: Share Transaction (previously described).

^OU: Abort Command (same as ^U).

^OW: Select Window Up; i.e., make the window above the current window the new current window.

^OX: Exclusive Transaction (previously described).

^OY: Close Window (previously described).

^OZ: Select Window Down; i.e., make the window below the current window the new current window.

^O1...^O9: Select Window 1...9 (previously described).

12.4. ^Q Prefix Commands. These are the commands accessed <u>after</u> typing Ctrl-Q. It is not necessary to hold the CTRL key down when typing the second letter of the command. It will always be interpreted as a control character.

^QA: Find and Replace (previously described).

^QC: Position at Bottom of File (previously described).

^QD or End: Position at End of Row (previously described).

^QE: Clear Window (previously described).

^QF: Find String (previously described).

^QG: Add Column (previously described).

^QH: Change Column (previously described).

^QI: Toggle Auto-Tab Mode; i.e., if Auto-Tab Mode is on (an "A" is on the Status Line) then turn it off. If it's off, then turn it on.

^QJ: Align Column (previously described).

^QK: Drop Column (previously described).

^QL: Keyword Search (previously described).

^QN: Select (previously described).

^QO: Join (previously described).

^QP: Project (previously described).

^QQ: Clear Query (previously described).

^QR: Position at Top of File (previously described).

^QS: Position at Beginning of Row (previously described).

^QT: Switch Columns (previously described).

^QU: Abort Command (same as ^U).

^QV: Sort Relation (previously described).

^QW: Unique (previously described).

^QY: Delete to End of Row; i.e., delete all data to the right of the cursor.

^QZ: Closure (previously described).

13. MACRO COMMANDS

There is a very primitive macro processor which is activated by F9. It does not allow any parameter substitution but merely processes the contents of the file as a series of keystrokes. To create a macro, simply type the sequence of keystrokes as an ASCII text file using capital letters to represent control characters. This can be done from within the system by using the Write File command to create an ASCII file.

14. INSTALLATION

14.1. The system requires a PC or 100% compatible with MSDOS 2.x or higher. Most of the popular video boards (Mono, CGA, Hercules, Paradise, EGA) are supported but the system does direct video output so it expects screen memory to be at $B800 for color or $B000 for monochrome.

14.2. The system will operate with as little as 128K but some functions will fail due to a lack of dynamic memory. It is fully functional at 256K but will make use all available memory up to 640K for buffer caching.

14.3. The system doesn't care much about directory structure and it supports path names including ..\ notation. However, the files TT.COM, TT.000 and TT.001 must be in the same directory and that directory is the default directory for all files unless a path name is specified. User file names may not have extensions as the system assigns its own extensions (.DEF, .DES, .TXT, and .X##).

14.4. It is possible to operate from floppies but the system makes very heavy use of overlays so execution is extremely slow.

15. REFERENCES

[1]. Conklin, J., "Hypertext: An Introduction and Survey," in Computer, IEEE Press, September 1987, pp. 17-41.

[2]. Queries execute off a stack which is particularly important to know when executing joins. A join is specified by an expression such as 1.TASK = 2.TASK. This defines a join on window #2 such that the top row in window #2 will have its TASK equal to whatever the TASK is on the current row (the row the cursor is on) in Window #1. However, the synchronization is not continual but only occurs when a ^L is executed in Window #1. For example, if a Select is defined in Window #1, then each time a ^L is executed, the next row qualified by the Select Expression is found, and the Join from Window #2 to Window #1 is re-executed. But it is not necessary that any query be defined in Window #1. A ^L will always cause the stack to execute. I.e., as long as the join is defined, you can simply position the cursor in Window #1 and issue a ^L command and Window #2 will be repositioned. Furthermore, the stack can be of any practical depth such as joining #2 to #1, #3 to #2 and #4 to #3 and #5 to #3. This creates a tree of joins with #1 as the root. Henceforth, any- time a ^L is issued in Window #1, #2 will be repositioned. Then, because #3 is joined to #2, it will be repositioned, etc., until all the joins in the stack have been executed (except circular joins, which will only execute once).

[3]. As of this writing, the UNION command does not work properly and is still undergoing tests. However, a Union can be accomplished by creating a Scan of all the rows in one table and then reading that scan into the other table. See the paragraph on Scans below.

[4]. As of this writing the Difference command is not working properly and is still undergoing tests. However, it can be accomplished by performing a Join using the "s" (scan) option, and then deleting the scan and writing what is left to another window. I.e., only the rows that did not join will remain which is the desired result (this must be done under transaction management or the original table will be corrupted.)

[5]. As of this writing, JSpaces have not been installed in the system. The programs to support JSpaces have been written and tested and are ready for installation, however the Phase I investigation has not yet produced enough data from which to create a JSpace, nor has it revealed a need for JSpaces in the scheduling problem. This may be due to the fact that the terminology of scheduling is fairly compact and standardized. If that remains the case, the system's existing support for synonyms will probably suffice for the Phase II effort. A brief discussion of JSpaces is provided in the section entitled "Related Work."

[6]. Formally such "queries" are considered to be beyond the power of first-order logic in that the quantification of the result after i iterations is dependent on the result of iteration (i - 1). In this sense, they are motivated (fueled) by the results produced along the way (and, perhaps, this is one justification for referring to them as "engines").

[7]. The technical requirements of these are described in the Objectives and Statement of Work sections of this proposal. They are redescribed here from a user's perspective. In the technical discussions, however, they are referred to as extensions to a single inference engine. Experience with the Phase I prototype, however, suggests that from a user's standpoint, a modular implementation of several engines is advantageous in two regards: (a) it retains the closed nature of the system in that every operation produces an object (table, scan, etc.) that can be manipulated by all the other functions of the system; and (b) a user has the flexibility to combine the engines in a number of ways, some of which may not be obvious even to the system designers.

[8]. Given a relation R(...,A,...,B,...) having two attributes over a common domain, it is then possible to have joins of indefinite length: R [A=B] R [A=B] R [A=B]... and in the general case the topological structure defined on the tubles of R is a digraph. We can invent a notation within relational algebra such that R^n[A=B] means R [A=B] R ... [A=B] R where R occurs n times. But there is no way to allow this to occur an indefinite number of times within relational algebra. We define the Closure operator R^+[K,A,B] as follows where K is a key of R, and A and B are

attributes or lists of attributes having the same domain:

> $R^{+}[K,A,B]$ is the set of all tuples $\langle K_1, K_2, n \rangle$ where there is a join sequence on A=B of length exactly n.

The Closure operator combines with other operators of relational algebra in such a way that it may occur on any semantic loop definable within the schema (e.g., upon a projection of a compound expression formed of joins). It has great import for part-whole relations such as determining the lowest common superior of a set of elements of a hierarchy, or the highest discriminants. For example, in the following hierarchy, the commonality between Algeria and Uganda is Africa, and the highest discriminants are (North Africa, East Africa).

1. Africa
    - 1.1. North Africa
        - 1.1.1. Mediterranean
            - 1.1.1.1. Algeria
            - 1.1.1.2. Libya
        - 1.1.2. Atlantic
        - 1.1.3. Interior
    - 1.2. West Africa
    - 1.3. South Africa
    - 1.4. East Africa
        - 1.4.1. Kenya
        - 1.4.2. Uganda
        - 1.4.3. Tanzania

Where the structure is a lattice rather than just a hierarchy, closure can determine the common inferior and its discriminants.