(2)

PL-TR-92-2097

A SINGLE-STATION FORECASTING MODEL

William H. Jasperson
David E. Venne
Miriam R. Peterson

AD-A256 403

Augsburg College
Center for Atmospheric and Space Sciences
731 21st Ave South
Minneapolis, MN 55454

1 April 1992

DTIC
ELECTE
JUL 15 1992
S A D

Final Report
15 December 1988 - 15 January 1992

Approved for public release; distribution unlimited

Phillips Laboratory
Air Force Systems Command
Hanscom Air Force Base, Massachusetts 01731-5000

92-18198

425636

This technical report has been reviewed and is approved for publication.

ROSEMARY M. DYER
Contract Manager

DONALD A. CHISHOLM
Chief, Atmospheric Prediction Branch
Atmospheric Sciences Division

ROBERT A. McCLATCHEY
Director, Atmospheric Sciences Division

This document has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/TSI, Hanscom AFB, MA 01731-5000. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
1 April 1992

3. REPORT TYPE AND DATES COVERED
Scientific Final (15 Dec 88---15 Jan 92)

4. TITLE AND SUBTITLE
A Single-Station Forecasting Model

5. FUNDING NUMBERS
PE 62101F
PR 6670 TA 10 WU DD

Contract F19628-89-C-0015

6. AUTHOR(S)
William H. Jasperson
David E. Venne
Miriam R. Peterson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Augsburg College
Center for Atmospheric & Space Sciences
731 21st Avenue, South
Minneapolis, MN 55454

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Phillips Laboratory
Hanscom AFB, MA 01731-5000

Contract Manager: Rosemary Dyer/GPAP

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
PL-TR-92-2097

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release;
Distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This report describes the development of an advanced prototype expert system, named Itasca, for the classical single-station weather forecasting problem. The forecasting skills necessary to produce forecasts from limited data are being lost though they continue to have application to certain military scenarios which include communications interruptions. Itasca is intended for use at an arbitrary mid-latitude station, and utilizes physical relationships between the synoptic weather and variables that affect the local weather rather than station specific rules. Itasca is designed as an imbedded expert system where the user has control over the operation of the system through a user interface. Rule bases are loaded and used as appropriate to the task at hand. The rule bases and the class/object structure used in the expert system were developed using the NEXPERT-Object development system. Testing of the system during development was limited by the lack of suitable datasets, but a plan is in place for further independent testing and evaluation.

14. SUBJECT TERMS
Weather Forecasting, Single-Station Forecasting, Expert System, Artificial Intelligence

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

TABLE OF CONTENTS

Accesion For

NTIS CRA&I
DTIC TAB
Unannounced
Justification

By
Distribution /

Availability Codes

Dist | Avail and / or Special
A-1

1. INTRODUCTION

The needs of the Armed Services for accurate, timely forecasts of environmental conditions have become increasingly critical with the evolution of current operational requirements and sophisticated weapons systems. Accurate forecasts of basic meteorological fields such as wind, temperature, cloud cover, precipitation, and visibility are essential for the planning, execution and success of almost any operation.

The best forecasts are created by a human forecaster, experienced with the geographic forecast area, interpreting data and computer-generated (numerical or statistical) forecast products. However, the Armed Services must be prepared to make critical forecasts even in the event that data and/or products from central collectors become unavailable. Questions that must be asked are: Is a forecaster trained to forecast the weather utilizing modern tools able to perform adequately when these tools are not available? How does a forecaster inexperienced with the forecast area perform when only limited observational data are available? Can a forecaster cope with both the loss of standard products and the pressure to provide useful forecasts in this scenario?

Artificial intelligence (AI) has been identified as a potential tool for solving difficult problems of this nature. Practical outgrowths of this field of research are knowledge-based software systems, sometimes referred to as expert systems. Knowledge-based systems have the capability to encapsulate the knowledge experts use to solve problems and make it useful to less experienced individuals.

During the past several years, several meteorological expert systems have been developed. The applications range from forecasting snowfall in Colorado (Swetnam and Dombroski[1]), to forecasting of thunderstorms and severe weather (Riese and Zubrick;[2] Elio, de Hann and Strong[3]; McArthur, R. C., J. R. Davis and D. Reynolds[4]), to interpretation of Doppler imagery in detecting wind shear hazards (Campbell;[5] Olson[6]; Campbell and Olson[7]), to forecasting fog (Stunder, et al.[8]). This last system has been evaluated for its applicability to locations other than those for which it was developed (Dyer and Freeman[9]). There has also been significant recent interest in evaluating different expert systems designed for forecasting severe weather, and the results of the most comprehensive comparison is given by Moninger, et al.[10]

The objective of the project described in this report was to develop a knowledge-based system, hereafter referred to as Itasca, that can be used to make short-range weather forecasts from limited input data. This "single-station" method of forecasting was developed during World War II for short range weather forecasting[11]. Specifically, Itasca is to make hourly forecasts of each of the input variables for the six to twelve hour forecast period. To our knowledge, it is the first system that attempts to forecast more than a single meteorological event. Providing a forecast of many variables implies a complexity that impacts the structure of the system, the user interface, and the data and knowledge contained within the system. This report describes the structure and operation of Itasca and concludes with comments and recommendations for further work.

2. SYSTEM DESCRIPTION

In the early stages of Itasca development, it was realized that the task of forecasting short-range weather from limited data requires a variety of skills. Some of these skills, such as heuristics concerning physical behavior, can be efficiently described in terms of collections of rules, commonly referred to as knowledge bases (KBs). Other skills may require the use of computations or sophisticated statistical methods.

The diverse set of tools required to embody these skills and make them accessible to the user suggested that the Itasca system could be most successful if it were a conglomerate of software tools. The alternative, in which the expert system is developed using a single tool or programming language, was judged to lack the flexibility needed to address the forecasting problem. Many previous systems built using only a single software tool (e.g., an expert system shell) or programming language (particularly one considered to be an "AI language" such as Lisp) have been very good at dealing with one aspect of their targeted problem but performed other functions poorly or inefficiently.

Therefore a primary design goal for Itasca was to give it the potential to use a wide variety of tools effectively and efficiently. This was accomplished by using a programming language (C) common to microcomputers and UNIX-based workstations as the "backbone" of the system on which the necessary tools could be supported. The added benefit of using a common programming language was that a variety of graphics and interfacing software packages were available for use. A description of the C program in which the control structure of Itasca is embodied is presented in Section 2.1.

Itasca currently uses a knowledge base system development tool called NEXPERT Object (hereafter referred to as NO) to create and maintain expert knowledge. NO provides a rich and powerful environment for the encapsulation and use of meteorological expert knowledge that is accessible from C language routines. NO knowledge bases are hierarchical, can be moved in and out of machine memory as needed and provide a freedom in knowledge representation that is advantageous during development of a system like Itasca. The NO knowledge bases, their organization and use are described in Section 2.2.

The process of meteorological analysis and forecasting is a highly visual exercise, therefore Itasca was designed to incorporate graphic capabilities. While the graphical display of meteorological features (for example, the geographic depiction of the locations of pressure centers and fronts) is a future goal for Itasca, the current user interfaces incorporate graphical as well as numerical representations of the observations. The Itasca interfaces are described in Section 2.3

2.1 Control Structure

The portion of Itasca that is coded in the C language is intended to perform the functions of analysis/forecasting session control, observation data management, graphics operations, user interactions and computationally-intense calculations. The first of these responsibilities, control, is delegated to the C programs because of the difficulty traditional knowledge-based systems have in expressing control operations. Small knowledge-based systems may not suffer significant penalties caused by the use of rules (or their analogs) to control system operations. However, large systems such as Itasca often have a significant portion of their rules dedicated to controlling operations that have little to do with expert knowledge manipulation. Routines coded in C are well-suited to controlling administrative and procedural operations; using C routines in this way allows the knowledge bases to be structured more clearly and concisely.

The controlling C routines are loosely divided into blocks that are roughly analogous to the major tasks of the forecaster. The initialization block establishes the local environment in terms of geography and climate. The observation block is used to input, modify, and preprocess surface and upper air observations. The analysis block produces a representation of the current meteorological state, and the forecast block is concerned with forecasting the future meteorological state. Some routines have wider applications and can be considered global rather than belonging to any one block. Examples of these global routines are computational routines and graphics and system utilities.

The management of observational databases is another important task of the C program. Itasca maintains a database for each type of observation (standard and special surface observations, radiosonde and pibal observations) and a number of routines allow the user to enter, inspect, modify, and extract observation data from these databases. Other routines search the databases for data to be used in making analyses and forecasts, in effect selecting a subset of observations that is relevant to the task. A special set of routines manages information about clouds, including those observed, inferred and potentially existing.

Meteorological numerical calculations are performed by a library of C routines. These routines contain the algorithms needed to determine such quantities as stability indices, thermodynamic values, diurnal variations, solar parameters, shears, time averages, and so on.

C routines are used to control all user interfaces and graphical presentations. These will be described in Section 2.4.

2.2 Knowledge Bases

2.2.1 Introduction to Terminology

Before describing the knowledge bases constructed during Itasca development, a brief overview of the terminology used in this Section may prove helpful. Although some of the terms used here are common to most traditional AI systems, the use of a specific knowledge base tool (in this case, NO) brings about an inevitable use of jargon. In this report an attempt is made to minimize the use of obscure or tool-specific words.

The primary knowledge base building block is a rule. Rules express knowledge that relates a condition to a consequence. Rules are often called "if-then" constructs because of the form of this relationship: *If* a condition or set of conditions is true, *then* the consequence is true. A simple example of a rule is *if* the 500 mb wind is from the west *and if* the observation is from the Northern Hemisphere, *then* low pressure is located toward the north. In NO the consequence is referred to as the *hypothesis*.

A condition in a rule may be (or contain) the hypothesis of one or more other rules. Therefore, to find the validity of a given hypothesis, other hypotheses may have to be evaluated. This linkage of rules through their conditions and hypotheses forms the basis of rule-based systems, and the rules can be evaluated in what is called backward and/or forward chaining. NO allows more complex methods of working with rules that will not be described here.

A set of *actions* may also be associated with each rule such that when a hypothesis is determined to be true (because all of its conditions are true), these actions will be initiated. These actions may be of many forms including initiating the evaluation of a new hypothesis, setting a value or evaluating a function.

Representational structures such as *objects*, *properties* and *classes* are used within an expert system to describe the entities in the system. An object in NO is just what its name suggests - an entity with one or more properties. In Itasca, the largest and most visible use of objects is the representation of meteorological features. A cold front is an example of an object within Itasca. While all cold front objects have the same set of properties, the values of those properties are different and specific to each individual cold front. Properties associated with an object are called *slots* (the terms "property" and "slot" will be used interchangeably). Slots can have numeric, text and special values such as Unknown and NotKnown. A slot has the value Unknown if the system has not tried to determine its value, and has the value NotKnown if the system has been unsuccessful in its attempt to determine the value.

A class is a grouping of objects that have one or more properties in common. For example, the class *Cold_front* defines the set of properties that all the cold front objects will have. When a particular cold front object is created, it has the ability to inherit the properties of its parent class, *Cold_front*. An object may descend from more than one parent class. The network of parent classes and child objects form a knowledge structure for representing complex entities within the system.

The purpose of rules in Itasca is primarily to create and destroy objects, when appropriate, and to determine the values of their slots. Slots may obtain values in several ways. One way is to inherit the value of the same property from its parent class. A second way slots are filled is through the actions of rules as previously mentioned. Another very useful way slots obtain values is through a set of actions that can be attached to the slot; these actions are executed when the value of the slot is requested but is currently unknown. These are called *order of source* actions, and may include the evaluation of the slot from an external source such as a function, the loading and execution of another knowledge base to determine the slot value, the querying of the user or the assignment of a default value.

Finally, the rule-based reasoning system may be linked to the object representation through an *if change* mechanism. Each slot of an object or class can define additional actions that are executed whenever the value of the slot changes. The possible actions are the same as described above and include loading a knowledge base, executing an external function, etc. Like slot values, order of source and if change actions can be inherited from parent classes or objects.

2.2.2 Itasca Class/Object Structure

The Itasca Class/Object structure is made up primarily of meteorological and geographic entities. The complete structure is given in the Appendix, but an overview and examples will be described in this section.

In Itasca, most entities within the class/object structure have been grouped using the geometrical descriptions of point, line or region. The point can be used to represent, for example, an observing station or the center of a high or low pressure center. A line can be used to represent features such as a mountain range or a meteorological front. A region can be used to represent objects with areal extent such as a body of water or an airmass. These three geometric descriptions are given properties of absolute and/or relative position. Points have the properties of latitude, longitude and elevation as well as direction and distance from some defined reference point. Lines are given the properties of orientation and direction and distance. Regions have properties of area, center latitude and longitude, and direction and distance from a reference point.

In addition to their geometric description, most objects maintained within Itasca are characterized by either time-dependent or time-independent behavior. In the examples above, the station, the mountain range and the water body are time-independent while the pressure centers, fronts and airmasses have time-dependent behavior. Therefore, two high level classes have been defined called *Time_dependent_obj* and *Time_independent_obj*. Objects created from subclasses of *Time_dependent_obj* inherit two slots: age and timestep. The age slot of an object contains the value of the number of hours that have passed since its creation. A number representing the current date and time (called a *timestamp*) is assigned to the timestep slot each hour. Attached to the timestep slot is a list of *if change* actions designed to express or trigger the time-dependent behavior of the object.

A moving geometric entity is a subclass of both the *Time_dependent_obj* class and the appropriate geometric class. It inherits properties from each of its parent classes, and it may be given additional properties of its own. For example, the class *Moving_line* inherits the properties of its parent classes *Time_dependent_obj* and *Line*, but it is also given properties characteristic of a moving line such as direction of motion (dir of motion), speed (speed) and the time that it passed or is expected to pass the forecast station relative to the current time (time to arrival).

All meteorological front classes (*Warm_front*, *Cold_front*, *Stationary_front* and *Occluded_front*) descend from the class *Front* that is a subclass of Moving_line and inherits all of the parent's properties. In addition, the *Front* class has properties such as a slope of the frontal surface and the height of the frontal surface above the ground. One distinction between front classes is the addition of the slot lowl name containing the name of the low associated with warm and cold fronts.

Other meteorological entities such as pressure systems and airmasses have analogous structures which can be found in the Appendix.

Another class defined in Itasca is the *Ob* or observation class. This class has a parent class *Times* which has properties representing the year, month, day, hour, minute and second. Subclasses of the class *Ob* are surface observations (*Sfc_ob*), upper-air observations (*Upa_ob*) and pibal observations (*Pibal_ob*). Also included but not utilized are subclasses for radar observations (*Radar_ob*) and satellite observations (*Satellite_ob*).

Each of these observation classes have appropriate slots defined for the types of observations they represent. For example, the class *Sfc_ob* has slots for all of the observed surface variables: temperature and dew point; pressure; wind speed, direction and gusts; up to four levels of clouds by amount, type and height and whether the height was measured or estimated; total and opaque sky covers; visibility; weather, remarks, and a sky and ceiling description; rainfall or snowfall and its water equivalent; snow depth. There are also some slots assigned to the surface observation class whose values are computed or determined from a rule base. These include items such as the pressure adjusted for diurnal change, the airmass type, the dew point depression, the 1-, 3-, and 6-hour pressure changes, and the presence of local circulations such as a land- or sea-breeze or drainage flow.

The class *Upa_ob* has slots for pressure, height, temperature, dew point, wind speed and direction for each of the significant and mandatory levels. There are also slots for surface observations of wind, temperature, dew point and pressure, the height and pressure of the tropopause, the maximum wind, and a number of computed values such as the wind shear between the gradient and 700 mb levels, the low level advection, the mean temperature and dew point for the low, middle and high level of the sounding, precipitable water and all of the common stability indices computed from the sounding.

The class *Fcst_ob* is a similar to the *Ob* class in that is contains slots for all of the variables for which forecast values are determined. In addition, it contains slots for the time of the forecast, the number of hours between the forecast time and the analysis time, the names of the last and the next forecast, and whether or not a land- or sea-breeze is forecast.

The *Station* class, used to define objects representing observation stations, inherits properties from its parent classes of *Point*, *Local_climate* and *Local_terrain*. *Point* provides location information, while *Local_climate* and *Local_terrain* provide local climatological and terrain data.

The *Local_climate* class has properties for the mean temperature, the mean maximum and minimum temperature, the mean dew point, the mean monthly precipitation, and the diurnal temperature range. It also has properties for the normal start and end time for possible land- or sea breezes. The user is interrogated for these values if the system determines that a land- and/or sea-breeze may be possible. The *Station* class also has slots for the ICAO and the WMO station identifiers.

The *Local_terrain* class has slots for local conditions that might alter the reliability of the observed wind. The user is questioned at the start of a session as to the extent of these influences and the slots are filled accordingly.

Even though provision for cloud observations are contained within the *Ob* class, Itasca also defines a *Cloud* class to reduce the problem caused by the observation techniques used in cloud reporting. Clouds are reported in tenths of sky cover, and, for an overcast sky, the coverage should add up to ten tenths. To illustrate the difficulty this type of observation causes for forecasting, assume there is a solid altostratus cloud deck at 8000 feet and a layer of cumulus covering five tenths of the sky at 4000 ft. This would be reported as five tenths cumulus and five tenths altocumulus. If the cumulus layer disappears, the next observation would be ten tenths altocumulus. Even though there is no change in physical cloudiness of the altocumulus layer, there is an apparent increase of clouds due to the nature of surface observations. The *Cloud* class provides slots for "true" cloud amount as well as observed cloud amount. The true value is computed using the assumption of uniform cloud distribution. This technique also allows the retention of higher layers such as cirrus when the sky becomes overcast with lower clouds. The forecasting of each cloud type is made separately. This process is internal to the system, and the presentation of forecast clouds are made as if they had been observed from the ground.

2.2.3 Itasca Knowledge Base Structure

The function of KBs is to contain the expert knowledge obtained from a meteorologist with single-station forecasting skills. KB domains are generally broken along lines of the control blocks noted above, with the exception of one global KB that exists for the life of the forecasting session. Non-global KBs will be loaded when needed and unloaded when they have served their purpose, permitting a dynamic allocation of resources and more sharply defined task domains. Loads and unloads of KBs are performed primarily by actions in other KBs; a notable exception to this is when the C control routines load (unload) KBs at the session start (end) and the beginning (end) of the analysis and forecasting processes.

The human meteorological forecasting process that Itasca is modelled on begins with the collection and analysis of all pertinent data and the assimilation of these data into a diagnostic model of the current synoptic situation. This is perhaps the most important part of the forecasting process because without a good understanding of the synoptic structure, the forecast is likely to be deficient. This diagnostic model is used to project a consistent forecast of future meteorological events.

There are advantages to structuring the expert system in this two-step diagnostic-prognostic formulation. First, it fits the actual manner in which meteorologists approach the forecasting problem whether it be forecasting using only single-station data or forecasting using state-of-the-art computer models. This makes the system more understandable to the user. Secondly, rules can be developed independently between the two parts of the problem. This is important because the type of rules or logic used in the two parts is inherently different. The diagnostic or interpretive part must use knowledge in a cumulative sense. Evidence is accumulated as more data become available. The forecasting part must explicitly take time into account in making the forecast.

Finally, separating the forecasting process into a model-building and a forecasting section allows the users to include their own input at the appropriate place in the forecasting process. That is, they may modify the model, and then the forecast will be based on that best-estimate model.

The KBs of Itasca are structured in this same way. Logically, the KBs can be grouped into three distinct sets. The first set has as its sole member the global knowledge base that contains all elements that must be accessible during both the diagnostic and prognostic phases of the cycle. These global elements include primarily the overall class structure and the objects that have been created. The meteorological objects, such as pressure systems and fronts, contain information about their location and movement that is necessary to the forecasting process. There are also a small number of rules in the global KB that are common to both diagnostic and prognostic evaluation. Currently these rules are used to either determine the effect of cloud cover on the observed temperature change or to forecast the cloud cover effect on temperature.

The other two KB sets include objects and rules that are necessary to either the diagnostic or the prognostic portion of the process, but not both. Both of these sets are loaded and unloaded once during the diagnostic/prognostic cycle, although in the prognostic case specific KBs may be loaded and unloaded once for each hour of the forecast. Itasca requires hourly observations, and an analysis must be made each hour. Forecasts are made for each of the first six hours as well as for the ninth and twelfth hours after the analysis time. The user is not required to initiate a forecast after each analysis.

2.2.3.1 Analysis

The first action taken by Itasca upon the user's initiation of an analysis is to determine the value of a variable called the representative wind (designated as slot rwnd in the surface observation object). The representative wind is the surface wind direction that is representative of the large scale surface pressure field. During the first analysis cycle, the user is queried about the possibility of small scale wind flows that may be present due to local topography. The KB that determines rwnd may request other KBs if the station is in a geographical location where other local circulations such as a land- or sea-breeze are possible. If the system determines from the geography that such a circulation is possible, the user is requested to fill in appropriate times of occurrence.

The variable rwnd is important because it is used to determine the location and movement of pressure systems and the possibility of frontal passages. During the presence of a local circulation (a sea-breeze, for example), the value of rwnd remains the same, assuming that it represents the larger scale pressure field upon which a local circulation is superimposed. When there is no specific local circulation, rwnd takes on the value of the wind averaged over a period of time.

The successful operation of Itasca is dependent upon reliable measurements of the wind. The representative wind KB attempts to determine the validity of hour-to-hour wind changes by considering factors such as the presence of convective precipitation in the area, the pressure and pressure changes, the presence of fronts, the strength of the wind, etc. The system may not perform well, however, if the input wind values are subject to persistent unreliability due to observation or measurement error.

The second action taken by Itasca during the analysis is the determination or verification of the airmass present at the station. This determination is made primarily with respect to current values of temperature, dew point temperature and cloud cover relative to climatological values of temperature and moisture. The airmass type is placed in the slot amtype in the surface observation object.

The third action taken by Itasca is to insert the timestamp value into the slot timestep. This is done to force all time dependent objects to update their slot values, using the new hour of observations. The primary time dependent objects include the meteorological objects: pressure systems, fronts and airmasses.

The slots (see Appendix) are updated by applying rules which utilize present and past observations as well as slot values from other objects. The slots that are currently active in front objects are the time the front is expected to pass the station (time to arrival), and the front's speed, distance, orientation, direction from the station and direction of motion. The front's time to arrival and distance are automatically updated each hour, but if either of these values or speed is updated, they are all recomputed.

A prioritized set of rules is applied to each slot of each object of the given class. For example, the orientation of a cold front may be determined or updated from the gradient level to 700 mb shear vector, if an appropriate sounding is available. Otherwise, if the front has passed, the orientation might be estimated by examining the wind directions before and after frontal passage. Or if the front hasn't passed, the orientation might be estimated from the wind direction before the frontal passage.

High and low pressure systems are also updated similarly. Slots that are evaluated each hour include the time to arrival (the time of passage of the highest or lowest pressure at the station), the value of the highest or lowest pressure observed at the station, the direction to pressure system and its direction of motion.

The only slot for the airmass objects that is evaluated each hour is the lower level slot which accepts the value of the surface pressure.

Objects also may be created or destroyed during the analysis phase. In Itasca, fronts are tied to low pressure systems; a warm front and up to two cold fronts may be associated with a low pressure center. Because single-station forecasting allows little information to be inferred about an object (front or pressure system) once it is well past the station, most slot values are reset to Unknown and are not evaluated further. The object and its time of passage are retained until a new object of the same type has been created, after which the object is deleted.

At the conclusion of each analysis cycle, Itasca updates the set of cloud objects. Each cloud object represents a cloud layer and/or cloud type. As discussed before, the knowledge bases in Itasca depend upon true cloud amounts rather than observed cloud amounts. Therefore, a cloud object may or may not be contained in the current observation. Cloud objects that represent clouds in the observation have their slot source set to the value "seen", while cloud objects that are retained but are not in the current observation have their slot source set to "inferred". Cloud objects are retained within the system until there is reason to delete them. Cloud objects are deleted if they should be visible in the observation but aren't or if a current sounding indicates a lack of sufficient moisture in the layer.

2.2.3.2 Forecast

Forecasts of the different meteorological observations cannot be made independently. For example, changes in temperature are affected by cloud cover and vice versa. Because of the dependence between variables, forecasts are made for each hour using the prior hour's forecast as if it were a valid observation. This stepwise procedure allows for some measure of concurrence between variables during the forecast period.

The forecast KBs provide forecasts for the variables in the following order: wind direction, speed and gusts; temperature, dew point; pressure; clouds; weather; visibility. All of the variables are forecast using the analysis (location and movement of pressure systems, fronts and airmasses), the last observation, the last forecast, and, if appropriate, geographical information such as the presence of large bodies of water.

The wind direction, speed and gust forecasts are based on the movement of pressure systems and the passage of fronts. In addition, wind speed and wind gusts may have a diurnal component. A diurnal component in wind direction, with the exception of a possible sea-breeze circulation, is not taken into account at this time.

Temperature forecasts are also based upon the passage of fronts as well as the diurnal component, modified by cloud cover or air flow over large bodies of water. The diurnal component is adjusted by use of an amplitude factor determined by the amount and height of cloud cover. Dew point is assumed to be constant in an airmass but is modified by advection in the vicinity of fronts.

Pressure forecasts are based upon the movement of pressure systems and the passage of fronts.

Clouds are perhaps the most complex forecast variable. A starting set of forecast clouds are generated by the controlling C-program shell of Itasca. These are essentially the cloud objects carried forward from the analysis and include observed as well as inferred clouds. In addition, potential cloud objects are created at levels determined favorable from a recent sounding. The forecast knowledge base then operates on this set of cloud objects by modifying the slot values containing cloud height, amount and type, and/or by deleting the object at some point during the forecast period.

Cloud forecasts are dependent upon the type of cloud and the synoptic environment. With the exception of diurnal convective cumulus, clouds are forecast on the basis of persistence, trends and the presence of fronts. For example, cirrus cloud amounts observed in advance of a front are increased over time. Middle level and low level clouds in advance of a warm front are increased and lowered over time. Soundings are used to determine moisture advection to modify existing clouds or to form clouds (create additional cloud objects) not yet observed. Convective clouds are created and forecast based upon the surface temperature, dew point and convective temperature.

Precipitation is forecast based on the synoptic environment, the presence of moisture, and the atmospheric stability. The type of precipitation (snow, rain, or freezing rain) is determined by surface temperature and sounding data. Precipitation is currently forecast if proper conditions are met. No probabilistic measure is used at this time. Precipitation, therefore, is forecast more often than it is observed.

Visibility is determined on the basis of type and intensity of forecast precipitation and/or the forecast of fog. Fog is forecast on the basis of dew point depression and the presence of conditions conducive to the formation of fog (radiation, advection, and precipitation cooling).

2.3 User Interfaces

Itasca makes use of a fully mouse-driven, menued user interface that is similar to that used by many popular software programs available to the public. The system has been designed with this type of interface because it is relatively easy to learn and use, and lends itself to the highly graphical interactive displays built into Itasca. An Itasca User's Manual has been prepared that describes the complete use of Itasca and its interfaces.

The interfaces are highly dynamic to assist users in operating the system. Menu items are automatically enabled and disabled as the forecasting session passes from one operational phase to another or as the use of command options renders them relevant or irrelevant. For example, the menu item used to initiate the production of a forecast remains disabled until an analysis has been produced.

Displays commonly referred to as dialog boxes are used by the KBs to request the information concerning land/sea breezes and terrain effects mentioned previously. Informational messages, system advisories and warnings are presented to the user through the use of dialog boxes. Dialog boxes are also used as editors as explained below.

Itasca was designed not as a fully automated system but rather to take advantage of user inputs and knowledge. As a part of this design, graphical tools were provided to allow the inspection and modification of analyses and forecasts. Aside from the command screen from which the command structure of Itasca is controlled, Itasca interfaces may be broadly classified as *Viewers* and *Editors*. Viewers are used to aid the forecaster in understanding either the data or system products (analyses and forecasts). Editors provide the user with the capability of entering or changing both qualitative and quantitative information. The Itasca User's Manual includes complete instruction for the use of all viewers and editors.

Itasca currently has viewers for surface and upper air data that are capable of displaying the observations in both graphics and text form. The graphical display of surface data includes time series of temperature, dewpoint, pressure, wind (direction, speed and gusts), clouds (amount and height), weather and visibility. The upper air data display is a fully interactive skew-T (or Stuve) diagram that allows the user to explore one or two soundings in a number of ways. The display also provides a variety of values computed from the sounding(s), including thermodynamic values, common stability indices, and shears. Upper air data may also be presented in height-time diagrams.

Editors are used to enter, edit, save and delete observations in a straightforward way. Objects used in making forecasts (including airmasses, fronts, pressure centers, and water bodies) and their slots can be edited by using specially designed dialog boxes.

3. TESTING AND EVALUATION

At the time of this report, Itasca has not undergone formal testing or extensive evaluation because of time constraints and difficulties in obtaining data of sufficient quality and completeness. However, informal testing and evaluation is an essential and continuing process during knowledge base development. The data used for testing the system during development consisted of approximately three weeks of hourly surface and twice-daily upper-air data from Omaha NE and Atlantic City NJ for April 1989, and an additional month of data (August 1988) for Omaha. The April data were manually entered into the system from the original MF1-10A and MF1-10B surface weather observation forms. Much of the cloud data were reported at 3-hourly intervals and had to be converted to hourly observations by interpreting the sky and ceiling observations and making them consistent with the synoptic cloud observations. The August data for Omaha were obtained from the USAF Environmental Technical Applications Center (ETAC) on magnetic tape. These data were deficient in that several fields were missing, including remarks, wind gusts and cloud layer data at non-synoptic times. A program was written to interpolate cloud data, but was not entirely successful because of the random errors and inconsistencies contained in the raw data. Therefore some of the data were subjected to manual editing. Some of the upper-air data also required manual editing.

Most of the testing and evaluation of the system has been done on the analysis portion. In general, the system generates a reasonable synoptic description of the current meteorological conditions. Pressure systems and fronts are created and destroyed at appropriate times and move through the analysis in a relatively orderly manner. Nine fronts and troughs moved through Omaha during the April time period, and only a very weak trough, with virtually no pressure signature and a two hour wind change, was not found by Itasca. Eight fronts and troughs were created at Atlantic City for the April test case and at Omaha for the

August test case. Again, pressure systems and fronts were created and destroyed at appropriate times.

The Itasca analysis is relatively sensitive to wind data. Winds that vary back and forth over fifty degrees or more over a short period of time can cause pressure systems to be relocated, and they are not always repositioned to a location that would seem appropriate based on the winds after the direction has become more constant. This occurs because, at the current time, Itasca does not maintain an historical record of the synoptic description or frontal positions; therefore, changes in the position of meteorological features may not be accurately reversed if the original movement was dependent upon a short term fluctuation. This is particularly noticeable in the timing of cold frontal passages, which is closely tied to pressure and pressure variations.

It was anticipated that Itasca might not perform well with the August test case since much of the development was done using springtime data. The magnitude and nature of the pressure variations are substantially different in the summer. However, the system performed quite well using the August data. It created the proper number of lows and anticipated the arrival of cold fronts within about 6 hours on several occasions. Some fronts did pass the station when they hadn't been anticipated within the twelve hour forecast period.

In general, Itasca appears to create an appropriate synoptic description. There have been instances where the system has generated an unrealistic synoptic environment (e.g. three lows and no highs), although most of these problems have been corrected. If this does happen, the system must be restarted. The time chosen to start the system also will impact the analysis generated by the system. Best results are obtained if the system is started with data taken from an uncomplicated synoptic environment; just as a forecaster will have difficulty discerning the synoptic situation if winds are light and variable and pressure changes are slight, so will Itasca. If the system does not start properly and doesn't correct itself within several hours, it should be restarted from an earlier time.

The forecasting portion of the system has been subjected to less testing and evaluation, but, in general, the forecasts appear to be reasonable reflection of the synoptic description. This is particularly true for temperature, dew point temperature, wind direction and speed. Clouds are difficult to forecast on the basis of single-station observations, but the system shows some capability in this area. Precipitation and visibility are probably the most deficient forecast parameters at this time, but there is reason to believe that performance can be improved.

Precipitation is currently forecast to occur much more frequently than it really does because it is based upon the synoptic environment and limited data. Future development should concentrate upon determining the knowledge required to generate a probability assessment and a presentation format that can communicate the information to the user.

With the limited data available for testing and the time constraint of the work, there are portions of the system (rules) that have not been exhaustively tested. As data are made available and testing continues, these problems will become evident and corrections will be made.

4. CONCLUSIONS AND RECOMMENDATIONS

The problem of forecasting the weather during periods when normal data and numerical forecast products are unavailable is one which must be considered in contingency planning. The classic single-station forecasting methodology provides a basis for operations under this scenario as well as other situations where detailed local short-range forecasts are required.

This report describes a project whose objective was to use single-station concepts and develop a knowledge-based system to make short-range station forecasts of the observed meteorological variables assuming availability of no data other than that collected at the station.

The result of this project is a system that:

* uses embedded KBs (providing analysis and forecasting capabilities) inside a traditional control structure

* provides for interaction with the user

* provides database management

* provides computational and procedural support

The modular nature of the design makes the system versatile and easy to operate. Training time necessary to learn how to operate the system should be minimal. The system as it currently exists is considered an advanced prototype.

Recommendations for the system and future development include:

Testing and Evaluation: Because of problems in acquiring satisfactory detailed surface observations on media conducive for transferring large quantities of data, the system has had only rudimentary testing. However, the development is at a point where substantial testing is possible and necessary. The availability of complete, quality controlled data sets for evaluation is still a problem. Particularly limiting are cloud data of sufficient description. The data used for testing to date were hand-entered from standard MF1-10A and 10B forms. However, complete transcriptions of these data are not available on electronic media. It is recommended that datasets of detailed hourly surface observations be developed for testing systems developed for short range weather forecasting. Portions of these datasets could be distributed to developers and portions could be maintained as standard independent test sets.

Further evaluation and testing should concentrate on determining environments for which the system does not perform adequately. It is recommended that testing begin with more extensive evaluation for stations located in the Midwestern United States. Evaluation should be made for both the diagnostic as well as the prognostic components of the system. As performance for these stations is evaluated, other stations should be added to the testing data set and results compared to the earlier stations.

It is recommended that standards be developed for verifying the system forecasts. Itasca is a limited data forecast system, and it should be tested against comparable forecast processes such as persistence, climatology, or limited data statistical models. Comparison of results with human forecasters can only be made if data availability is the same. Documentation of the results of such testing can be used to improve the system performance.

An inherent limitation of Itasca is that it is designed such that a minimum amount of information is required of the user. If the system cannot determine the answer to a question, it assumes that there is no known answer and proceeds accordingly. It does not, in general, prompt the user for an answer. This is in accordance with the scenario of little available data and no experienced forecaster operating the system. However, answers to such questions as "Is the sky brightening?" are important to the short range weather forecast but are not often included in observation remarks. The system tries to overcome this limitation by allowing the user to modify the analysis on the basis of his/her knowledge, but the modification must be initiated by the user.

<u>Databases</u>: At the present time, there are only limited databases available to the system. The meteorological database currently contains average monthly maximum and minimum temperature from the World WeatherDisc[12], average diurnal wind speed variation and average diurnal pressure change which has been given a latitudinal variation. There

are a number of additional databases that would be useful. For example, databases of airmass source regions, average airmass characteristics as a function of latitude and longitude, storm tracks as a function of month or season and upper-air climatology would all be very useful.

The geographic database currently consists of map strings (latitude-longitude pairs) from which land and water bodies can be determined. However, there is no information on water body surface temperature which is very important in forecasting temperature, moisture, clouds and fog for stations in the vicinity of these water bodies. The geographic database also contains a topographic grid that contains terrain elevation and roughness parameters on a scale of 6 minutes, although these data are not used in any substantial way in the system.

Even if these data were easily available, the problem of determining how to efficiently and usefully represent this type of information within the system has not been solved. The disparity between the human ability to synthesize visual features and patterns and the ability to simulate this process on the computer is real, significant and important, and more work is necessary.

System Design: The most significant deficiency in the current design is that the system maintains no history of how things have evolved. The objects are created and destroyed as necessary, and they maintain information (slot values) about themselves. However, they know nothing about the sequence of how they all came to be. If those data are changed because of an unexpected change in the observations, which at the next hour is nullified, the system has no way of recovering well. This places a large dependency on accurate, representative observations. For example, a sudden large drop of pressure may mean that a frontal passage is imminent and the time to cold front passage may be reduced to 1 or 2 hours. If the next hour pressure change is small, or possibly even a rise, the system might be better off with the original value which had been replaced, and therefore lost. The objects within the system should retain a history of their attributes.

REFERENCES

1. Swetnam, G. E. and E. J. Dombroski, 1985: A Demonstration Expert System for Weather Forecasting. Preprints: 2nd International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology, Boston, Amer. Meteor. Soc., 310-316.

2. Zubrick, S. M. and C. E. Riese, 1985: An Expert System to Aid in Severe Thunderstorm Forecasting. Preprints: 14th Conference on Severe Local Storms, Boston, Amer. Meteor. Soc., 117-122.

3. Elio, R., J. de Haan and G. S. Strong, 1987: METEOR: An Artificial Intelligence System for Convective Storm Forecasting. J. Atmos. Ocean. Tech., 4, 29-35.

4. McArthur, R. C., J. R. Davis and D. Reynolds, 1987: Scenario-Driven Automatic Pattern Recognition in Nowcasting. J. Atmos. Ocean. Tech., 4, 19-28.

5. Campbell, D., 1986: Microburst Recognition: An Expert System. Preprints: 23rd Conference on Radar Meteorology, Boston, Amer. Meteor. Soc., 26-29.

6. Olson, S. H., 1986: Overview of an Artificial Intelligence System for Gust Front Recognition and Tracking. Preprints: 23rd Conference on Radar Meteorology, Boston, Amer. Meteor. Soc., 22-24.

7. Campbell, S. D. and S. H. Olson, 1987: Recognizing Low-Altitude Wind Shear Hazards from Doppler Weather Radar: An Artificial Intelligence Approach. J. Atmos. Ocean. Tech., 4, 5-18.

8. Stunder, M. J., R. C. Koch, R. N. Sletten and S. M. Lee, 1987: ZEUS: A Knowledge-Based Expert System that Assists in Predicting Visibility at Airbases. AFGL-TR-87-0019, Air Force Geophysics Laboratory, Hanscom Air Force Base, MA. (ADA184197)

9. Dyer, R. M. and G. L. Freeman, 1989: Rule-Based Systems for Visibility Forecasts. GL-TR-89-0116, Geophysics Laboratory, Hanscom Air Force Base, MA. (ADA214622)

10. Moninger, W. R., J. Bullas, B. de Lorenzis, E. Ellison, J. Flueck, J. C. McLeod, C. Lusk, P. D. Lampru, R. S. Phillips, W. F. Roberts, R. Shaw, T. R. Stewart, J. Weaver, K. C. Young and S. M. Zubrick, 1991: Shootout-89, A Comparative Evaluation of Knowledge-based Systems That Forecast Severe Weather. Bull. Amer. Meteor. Soc., 72, 1339-1354.

11. Oliver, V. J. and M. B. Oliver, 1945: Weather Analysis from Single-Station Data, Handbook of Meteorology, F. A. Berry, E. Bollay and N. R. Beers, eds., New York, McGraw Hill, 858, 879.

12. World WeatherDisc, WeatherDisc Associates, Inc., 4584 NE 89th, Seattle, WA 98115.

APPENDIX

Itasca Class/Object Structure

Key:

Class
properties defined by class
Subclass (inherits slots from **Class**)
properties defined by subclass

.
.
.

Most classes inherit from two main classes:
Time_dependent_obj and Time_independent_obj

Other classes that do not inherit from these two main classes are:

Cloud
Cloud_layer
Times
Local_climate
Local_terrain
Line
Point
Region
Wave_phenomena

Time_dependent_obj

- *age*
- *timestep*
- **Moving_line** (also inherits from Line; see below)
  - *dir_of_motion*
  - *speed*
  - *time_to_arrival*
  - **Front**
    - *height*
    - *slope*
    - **Warm_front**
      - *low1_name*
    - **Cold_front**
      - *low1_name*
    - **Stationary_front**
    - **Occluded_front**
  - **Moving_wave**
    - **Pressure_wave**
      - *slope*
      - **Pressure_ridge**
      - **Pressure_trof**
- **Moving_point** (also inherits from Point; see below)
  - *dir_of_motion*
  - *speed*
  - *time_to_arrival*
  - **Cyclone**
    - *cyc_type*
    - *low_name*
    - *occl_type*
  - **Pressure_center**
    - *central_pres*
    - *intensity*
    - *maxmin_pres*
    - *maxmin_obs_pres*
    - *maxmin_pres_tta*
    - **High_pres_center**
    - **Low_pres_center**
      - *cold_front1_name*
      - *cold_front2_name*
      - *cyclone_name*
      - *warm_front_name*
- **Moving_region** (also inherits from Region; see below)
  - *dir_of_motion*
  - *speed*
  - **Airmass**
    - *amtype*
    - *lower_level*
    - *upper_level*
    - **cP_airmass**
    - **mP_airmass**
    - **mT_airmass**
    - **cT_airmass**
    - **cA_airmass**
    - **modified_airmass**
    - **return_flow_airmass**

Time_independent_obj

    Geog_feature

            angle_subtended
            direction_closest
            distance_closest

        Hill (also inherits from Region; see below)
                orientation
                roughness

        Mountain (also inherits from Region; see below)
                orientation
                roughness

        Plain (also inherits from Region; see below)
                slope
                slope_direction

        Water_body (also inherits from Region; see below)
                direction_center *radial_size*
                distance_center *surface_temp*
                kind *surface_type*

Cloud

        age *height_trend*
        amount_obs *height_true*
        amount_trend *name*
        amount_true *source*
        creation_time *type*
        height_obs

    F_cloud
            trigger

    P_cloud
            thickness
            used

Cloud_layer

        layer_name

    High_cloud
    Low_cloud
    Middle_cloud
    Obscuration_cloud
    Vertical_cloud

Times

dayn *secs*
hour *timestamp*
mins *year*
mnth

Ob

Sfc_ob

amtype	*c4hc*	*rain*
aslp	*c4ht*	*rmrk*
clam	*c4ty*	*rwnd*
clhc	*cnum*	*sdep*
clht	*ddep*	*sea_breeze*
clty	*dewp*	*skyc*
c2am	*drain_flow*	*snow*
c2hc	*equi*	*temp*
c2ht	*land_breeze*	*tscv*
c2ty	*oscv*	*visi*
c3am	*pclh*	*wdir*
c3hc	*pc3h*	*wgst*
c3ht	*pc6h*	*wspd*
c3ty	*pres*	*wthr*
c4am		

Sp_sfc_ob

St_sfc_ob

Upa_ob

cclv	*max_dewp*	*temp0100-1000*
ddep0100-1000	*mwlv*	*temp_high*
dewp0100-1000	*pott0100-1000*	*temp_low*
dewp_high	*prec*	*temp_mid*
dewp_low	*pressfc*	*tempsfc*
dewp_mid	*stct*	*trop*
dewp_sfc	*stki*	*trph*
front_speed	*stli*	*wdir0100-1000*
grad700_shear_dir	*stsi*	*wdirsfc*
grad700_shear_spd	*stti*	*wmax*
hght0100-1000	*stvt*	*wspd0100-1000*
lclv	*stwi*	*wspdsfc*
low_level_advection		

Ra_ob

Pibal_ob

front_speed	*grad700_shear_spd*
grad700_shear_dir	*low_level_advection*

Radar_ob

Satellite_ob

Forecast

Fcst_ob

amtype	*c3ty*	*pres*
aslp	*c4am*	*sea_breeze*
c1am	*c4ht*	*temp*
c1ht	*c4ty*	*tscv*
c1ty	*dewp*	*valid_time*
c2am	*hrs_past_analysis*	*visi*
c2ht	*hrs_since_fcst*	*wdir*
c2ty	*last_fcst*	*wgst*
c3am	*land_breeze*	*wspd*
c3ht	*next_fcst*	*wthr*

Local_climate

dTrange	*meantemp*
land_breeze_end	*meanTmax*
land_breeze_possible	*meanTmin*
land_breeze_start	*sea_breeze_end*
meandewp	*sea_breeze_possible*
meanprec	*sea_breeze_start*

Station (also inherits from Point; see below)

icao
locn
wmon

Local_terrain

no_terrain_influence	*bad_wd2_2*
number_bad_wdirs	*drainage_flow_poss*
bad_wd1_1	*drain_flow_dir*
bad_wd1_2	*drain_flow_ws*
bad_wd2_1	

Station (also inherits from Point; see below)

icao
locn
wmon

Line

direction
distance
orientation

Moving_line

(see above, under Time_dependent_obj)

Point

direction	*lati*
distance	*lngi*
elev	

Moving_point

(see above, under Time_dependent_obj)

Station

(see above, under Local_climate)

Region

bounded_area *direction*
center_lati *distance*
center_lngi

Moving_region

(see above, under Time_dependent_obj)

Hill

Mountain

Plain

Water_body

(see above, under Time_independent_obj)

Wave_phenomena

amplitude
wavelength

Moving_wave

(see above, under Time_dependent_obj)