Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 508439
UNCLASSIFIED	

TITLE

GEM A GENERIC MODELLING UTILITY APPLIED TO HUMAN RESOURCE MANAGEMENT

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to: TC

DEPARTMENT OF NATIONAL DEFENCE

CANADA

OPERATIONAL RESEARCH AND ANALYSIS ESTABLISHMENT

DIRECTORATE OF MANPOWER ANALYSIS

D MAN A STAFF NOTE 8/91

GeM
A GENERIC MODELLING UTILITY
APPLIED TO HUMAN RESOURCE MANAGEMENT

by

P. BENDER
and
S. ISBRANDT

OTTAWA, CANADA

DECEMBER 1991

PREFACE

The contents of this paper have been included in the "Compendium on Futures and Forecasting Methods and Techniques for use in Government" published by the Interdepartmental Committee for Futures and Forecasting (ICFF).

TABLE OF CONTENTS

GeM
A GENERIC MODELLING UTILITY
APPLIED TO HUMAN RESOURCE MANAGEMENT

BACKGROUND

Necessity

1. In the Directorate of Manpower Analysis (D Man A), it was clear a few years ago that there was a requirement for a manpower career progression model which demonstrated immense flexibility for modifying scenarios. There is routinely a need to analyze new and different policy options which are difficult and time-consuming to incorporate into existing career progression simulations.

2. Requirements for streamlining updates, modifications and variations came to a fore with the requirement to support the Trade Advancement through Skill and Knowledge (TASK) impact analysis. A TASK career progression model would have to cater to a two dimensional career flow instead of the existing one dimensional flow, and would have to cater to structural differences for each occupation. In addition such a model would require vast flexibility for analysis of policy options not yet clearly defined. All of this would have to be accomplished with as much ease and timeliness as possible.

3. Examination of existing models led to the conclusion that extensive effort would be needed to alter them in order to provide the minimum functionality required. Extending such models, written in procedural code, to capture TASK would tend to create programming nightmares related to problems associated with arrays, data structures, and resulting subscript debugging. It was much simpler and more efficient to design a modelling utility from first principles, allowing for the development of new models from the conceptual level as opposed to the computer program code level.

Development Approach

4. Having determined that a new approach was required, an examination of tried and true as well as current state of the art techniques was undertaken. Existing models had been developed using computers and languages which were state of the art 10 to 15 years ago. However, the advent of powerful personal computers and affordable computing environments, coupled with increased understanding in knowledge-based and object-oriented techniques, allowed the investigation of completely different methodologies.

5. Of all the approaches considered, two of the most promising were examined in detail. The first approach was based on object-oriented programming, and the second centred around knowledge-based techniques. Both methodologies were evaluated using the idea of a generic "core" simulation driver to facilitate model building and modification.

6. The concept of a basic "core" simulation driver is presented in Figure 1. User defined data representing the scenario to be modelled are contained in three modules:

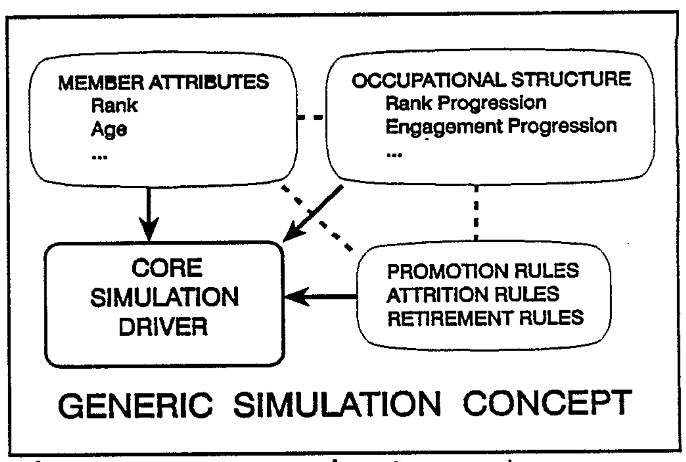

Figure 1 - Generic Concept

a. the set of member attributes (such as time in rank, years of service, etc) used by a simulation;

b. the occupational structure (for example sergeants are promoted to warrant officers); and

c. the various rules, policies, or attrition patterns appropriate for a human resource management simulation (for example, the Compulsory Retirement Age for the military).

7. Interactions between these modules are represented by dotted lines in Figure 1. The user defined modules must be developed in a consistent manner to ensure integrity of the simulation. The core simulation driver coordinates the information from the three modules into a coherent and integrated whole that represents the model scenario.

Prototypes

8. The knowledge-based prototype is referred to as MANSKEL, for Manpower Skeleton model (Figure 2). Member information and model rules provide input to an inference engine, where knowledge-based techniques are applied. Rules representing Human Resource Management (HRM) policies and Canadian Forces (CF) career progression are applied to the existing fact base repetitively until no new facts are obtained. This resolution of current facts with the rules provides new facts. (Who is eligible for promotion, who has reached Compulsory Retirement Age, et cetera). These new facts are acted upon by one or more post processors, which perform bookkeeping functions associated with attrition, retirement, promotion and so on. This produces a subsequent new fact base. The model increments specified attributes (one more year of service, one more year time in rank, etc. for members). The

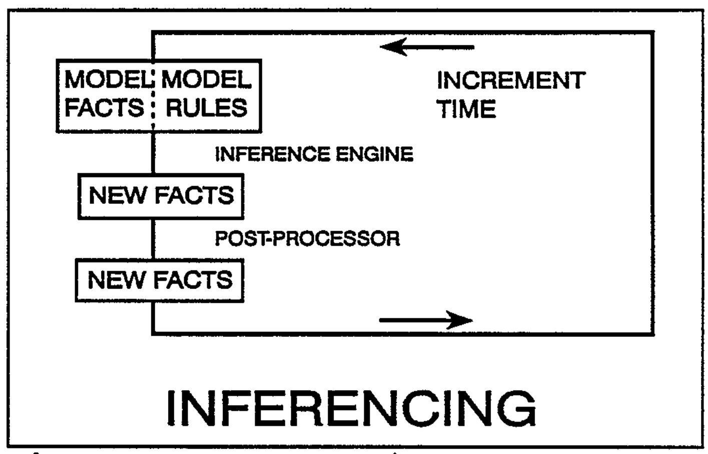

Figure 2 - Inferencing Model

updated fact base is used as input for the next simulated year, and the cycle repeats itself until the simulation is complete.

9. In the object-oriented prototype the "core" simulation driver consists of a Simulation object and its associated methods (routines). Outside of the driver, the member attributes are taken care of by the Member object and its methods. The occupational structure and the various rules correspond to the Occupation and Trade Cell objects and their methods. An execution of the model first creates a Simulation object. Simulation methods in turn create an Occupation object, whose characteristics are input by the user. The structure of the occupation is used to define a number of Trade Cell objects, corresponding to the rank qualification pairs. The Simulation reads member attribute information, creates Member instances corresponding to individuals in the occupation being modelled, and places them into the member list of the appropriate Trade Cell. Simulation of career progression then proceeds, calling methods to apply rules for attrition, promotion, bringing in recruits, and so on. An update method carries out necessary bookkeeping (increasing age, years of service, time in rank, and such) before cycling the simulation for another year. When changing model parameters, except for a few bottom-level methods, the core simulation driver would normally not be touched.

Specifications

10. Proof of concept prototypes were built for both approaches, and compared. It was decided that the vastly increased flexibility sought could best be attained with the knowledge-based approach, although it was clear that the object-oriented approach had worthwhile features. The MANSKEL knowledge-based prototype, although not practical for running large real-life problems, was entirely successful as a proof of concept.

11. MANSKEL was the inspiration for specifications subsequently created. Eventually object-oriented features were incorporated in the final product, resulting in a blend of the best of the two approaches. Suitable specifications were created for a modelling tool to handle the large problems which D Man A encounters. The model requirements were significantly different from previous contract experiences. Accordingly, the specifications were developed in conjunction with a contractor who specialized in the design of specifications leading to a Request For Proposal (RFP). These specifications emphasized the functionality of the modelling tool to be developed, based on experience with the prototypes. The eventual developer was not to be constrained by predetermined choices of particular detailed techniques, programming language, etc. The specifications endeavoured to define the capability of the modelling utility. Potential contractors were left to determine how they could best apply their expertise and knowledge to provide the required functionality, and to explain how they would do so. The experience of prototype development significantly facilitated both the design of specifications, and the evaluation of contractor proposals. Eventually, a contract for development was awarded, and the new modelling utility was obtained at the end of March 1991.

GENERIC MODELLING (GeM) UTILITY

12. The utility is referred to as the Generic Modelling utility (GeM). Although designed for HRM simulation to assist with the assessment of actual or hypothetical policies applied to occupations within the CF, GeM provides desired flexibility along with a high degree of generality. The following brief description will present it at the conceptual level using examples taken from a military HRM scenario, although it is also suitable for more general applications.

Overview

13. GeM is actually a model building utility. It provides for a variety of constructs that are coordinated by a user through a series of menus. It assumes that a simulation process takes place over a number of time slices, and provides features which can be used to control the sequencing of the simulation elements within a time slice.

14. The GeM utility is capable of simulating a broad range of scenarios characterised by state transition processes in a discrete environment. Although the utility was designed for developing and running HRM models it has proven to be suitable for modelling in other areas. GeM can be used in areas other than HRM. In particular, it could be used to develop prototypes that would demonstrate the strengths of the methodology. Successful prototypes could lead to GeM enhancements, resulting in a modelling utility suitable for other fields.

Use of Advanced Programming Techniques

15. The GeM utility incorporates the best of both approaches investigated in the prototypes. The heart of the utility is an Inference Engine (IE) that permits the alteration of individual attributes or model parameters based on complex matching criteria on the member attributes. The IE provides the utility with its immense flexibility and contributes in large part to the generic nature of GeM.

16. In addition to the IE the utility makes heavy use of object-oriented techniques. The object-oriented nature of the utility facilitates the creation and coordination of model objects. Whereas the IE is transparent to the user until the most powerful features of GeM are used, the object-oriented approach is evident from the very beginning of the model building process.

Model Structure

17. The typical layout of a model developed with GeM is represented at Figure 3. There are three portions to a model stored as an integrated text file. The middle portion is common to every model and consists of all the object class definitions and their associated methods, and is commonly referred to as the tool box. The initial portion of a model consists of its user defined aspects. It comprises the various objects created from the toolbox (links, nodes, etc) and associated information (parameters, rule bases, etc) that represent the model scenario. Finally, data on individuals (1 record per individual) makes up a database, that is the third and last portion of a model.

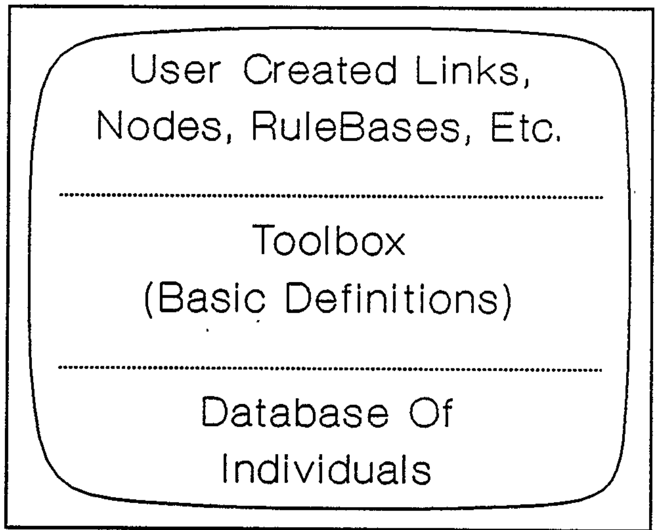

Figure 3 Model Structure

18. Input to GeM consists of information which defines model facts and rules, corresponding to career progression policies and structures, as well as an integrated database of attribute information for individuals. (An attribute is any characteristic used by the model and could include items such as age, rank, time in rank, etc).

19. Simulations, developed using GeM, model the flow of individual members of a population rather than aggregate stocks. This allows for the modelling of much more complex rules and policies than would be otherwise possible. In addition, small populations can be projected within a model more appropriately using Monte Carlo simulation since the problem of fractional individuals can be avoided. Modelling using aggregate stocks can be useful for large

populations in the analysis of major trends. Results from these models could be examined in further detail with the use of an individual based model. Consideration is being given to the development of a complementary stock and flow model incorporating the features of GeM.

Model Building Elements

20. The fundamental building blocks of GeM are referred to as links and nodes. Within GeM there exists a close association between them. Typically nodes characterize a state within the simulation and associated links define the various ways that individuals can be transformed to that state. By calling upon their intrinsic rule base, links perform attribute value transitions on individuals of a population. Internal node and link counters are dynamically updated to reflect the numbers resulting from the transformations.

Links

21. One of the basic model mechanisms is pictured in Figure 4 as an arrow. In the model terminology, it is referred to as a link, and performs the state transitions which are fundamental to any simulation. Links are the actual workhorses in any model developed from GeM.

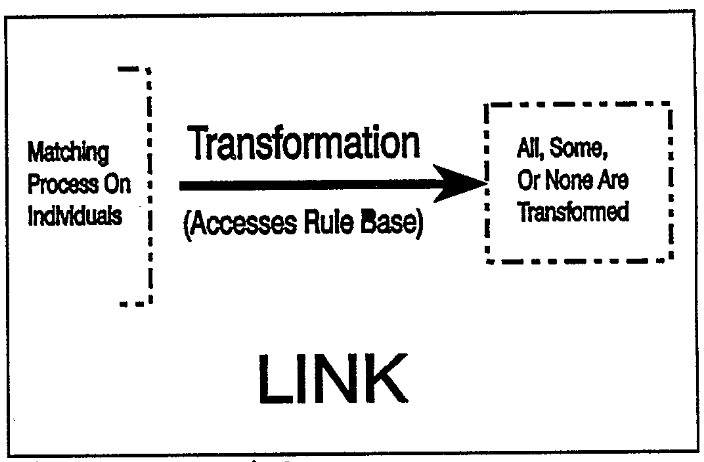

Figure 4 - Link

In general terms, links select individuals according to a matching process using a user defined template, and then may carry out a state transition for each individual match. Each template matches on attribute characteristics (eg a single value, membership in a range or a set of values) of individuals within the simulation. The number of transitions amongst the selected individuals can be limited by variables in both the node to which the link is attached and the link itself.

22. GeM provides for several types of links which include the standard link (std_link), the push_link, the age link, the exit link, the log_link and the entry link.

a. The **standard link** effects transitions but is subject to limitations imposed in the head node, while

b. a **push link** ignores such limitations.

c. The **age link** is the mechanism used to move the simulation one time step forward.

d. An **exit link** is used to remove individuals from the simulation environment by deleting them from the model database.

e. A **log link** is used to copy individual records to an external file for subsequent analysis or reporting purposes.

f. Finally, an **entry link** is used to generate new individuals in the simulation.

23. As stated previously, a link is used with a user defined template to match on attributes of individuals within the simulation. In one model, a template was used to match on all individuals having the attributes of: being a Sergeant, holding Qualification Level 6B and having no more than 29 years of service. Having matched on a number of individuals, a transformation can be carried out on all of them or on a subset. A transformation consists of changes to one or more of an individual's attribute values. For instance, if some of the Sergeants are promoted to Warrant Officer, the promotion is reflected with a change in the rank attribute value for those affected. GeM provides a

straightforward way to make simple updates to attributes of individuals who satisfy the matching process on a link.

24. A link can also be associated with rules where an inferencing process is carried out on the database representing individuals, facts, and model structure. This allows more complicated transaction processes to occur, rather than just simple alteration of attributes. The rules are created in an interpreter environment which corresponds to a basic subset of Clocksin and Mellish Prolog.

25. In spite of the fact that links are associated with nodes for the purpose of record keeping, nodes do not restrict the matching process. The links perform their matching process over ALL individuals in the database. This puts the onus on the model designer to make sure that the model representation is consistent with what is intended.

Nodes

26. Nodes characterize certain states by focusing on attribute characteristics of the population. They also control the sequence of events within a simulation. At the state transition level they control the order of the links that effect the transitions to the state represented by that node. Nodes are important in driving a simulation since they have an associated capacity that typically corresponds to the number of individuals desired at a certain state. As the number of individuals fall below this preferred number, the links tied to the state will attempt to transform sufficient numbers of individuals to satisfy the node requirements.

27. GeM differs from off-the-shelf simulation applications in its one to many mapping of individuals to states. Other applications tend to represent the state space as a set of mutually exclusive states where an individual is represented in only one of these states at any time. On the contrary, GeM allows individuals to be

represented in many distinct processes represented by different sets of states simultaneously; thereby permitting a more natural and conceptual way of representing a system, often with far fewer states than would be possible with other methodologies.

28. The various types of nodes available within GeM are referred to as the node, the age node and the export node. There is no difference between these various types other than specific link types must be connected to certain node types. As an example an exit link must be connected to an export node.

Model Development

29. A design of a new model must be prepared prior to using GeM. However designs in traditional computer languages do not make full use of efficiencies that can be gained using GeM. A big advantage of GeM is that model design tends to relate directly to the conceptual representation of the problem, and gives GeM a very different feel from other modelling methodologies.

30. The most simple method of model development using the GeM utility is to load an existing model; and then to delete unwanted portions and to add new links and nodes to reflect new features needed. On occasion it is necessary to build a new model from the basic building blocks through the menus. The core model only contains definitions of the basic building blocks and their associated primitives. Before creating links and nodes the user must define the set of attributes for each individual that will be modelled. In this way the user defines policies that relate to these attributes.

31. In general a model can be represented by diagrams of links and nodes where nodes typically represent the state that results from a transition effected through links from a different state. By connecting links and nodes together all associated counters are

dynamically updated. The general structure is represented with a simple example (Figure 5) which illustrates a simple career progression from private (pvt) through corporal (cpl) to sergeant (sgt).

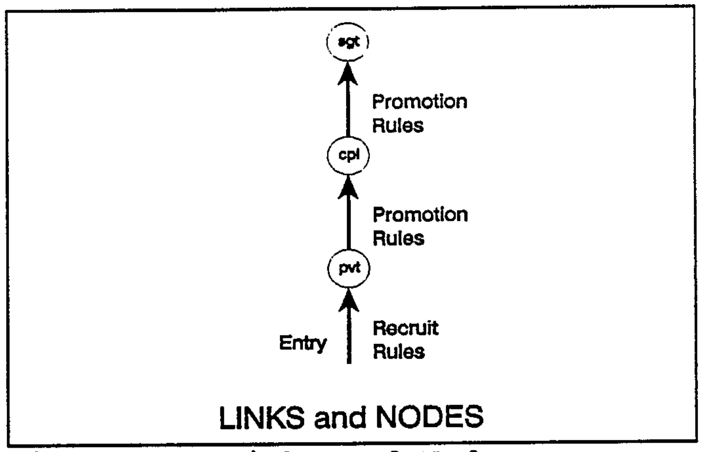

Figure 5 - Links and Nodes

<u>Use of Overlaid Nets</u>

32. One of the most powerful features of GeM is associated with the use of distinct overlaid networks or streams, possible since all individuals are contained in a dynamic database accessible to the whole model. Individuals are no longer uniquely identified with any one link or node; rather an individual can now be associated with links and nodes of two or more different streams simultaneously.

33. For instance, if a model were to keep track of two distinct (but not necessarily independent) processes, one based on rank, and one on engagement, they could be portrayed and modelled as two streams. In Figure 6, there are again the same rank nodes in one stream, and nodes representing four different engagement types in another stream. The two streams are distinct processes, but since individuals have both a rank value and an engagement type they will actually match and be counted in two nodes simultaneously (one match in each stream). The user must ensure that the Rank Stream does not explicitly change an

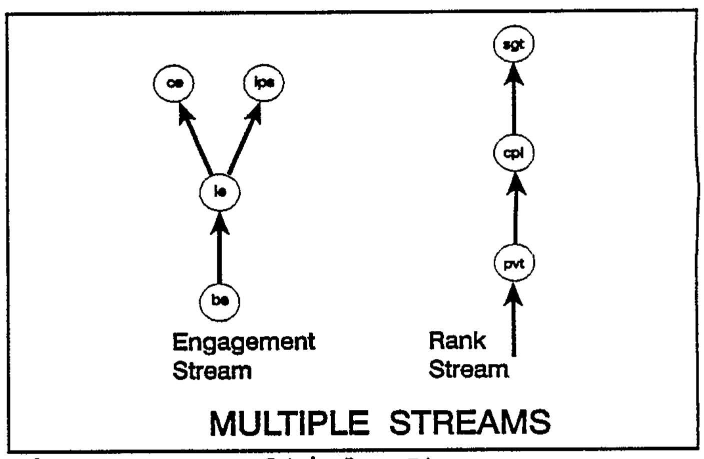

Figure 6 - Multiple Streams

Engagement attribute, and vice versa; and interactions between the streams can be handled through the rule bases. For example, if on promotion to sergeant (sgt) all individuals are to be offered the Indefinite Period of Service engagement (ips), then the rule base for the transformation link to ips could include matches on sergeants who have an intermediate engagement (ie). By sequencing engagement links after the rank links, updates will be carried out within one time slice as desired.

Interpreter Environment

34. The menus provide an easy way to create basic structures and associate them with basic model input. A large number of models, both simple and complex, can be created by using menus alone. When the menus no longer cater to more complicated data structures or complex interactions, the user can turn to the interpreter environment to continue development. Within this environment, which is a strict subset of Clocksin & Mellish Prolog, the user can develop complex rules or knowledge-based systems from first principles.

35. Rule sets can be built, saved, and associated with links through a rule interpreter, which is accessible through the same main user interface. Provision of rules associated with links can be viewed as multiple independent "expert systems" within a model framework. In Figure 7, rules designate the criteria for transformation to the ranks of private, corporal and sergeant. The approach provides increased flexibility within the model building utility. Changing the rule

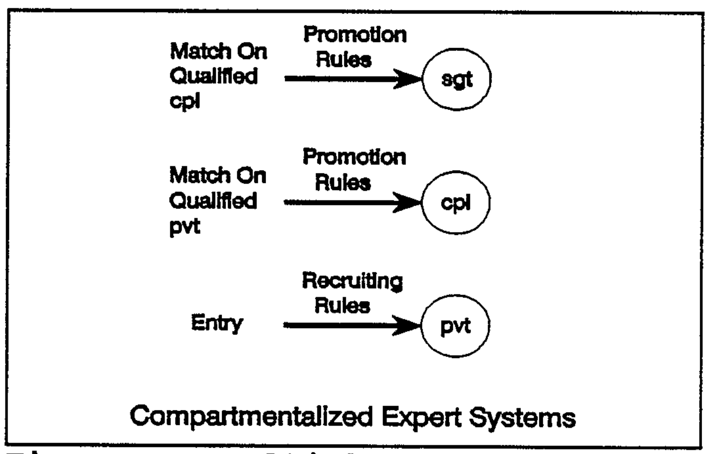

Figure 7 - Multiple Expert Systems

base attached to a link is, in cases where very significant model changes are made, considerably streamlined as compared to making a major change to a complicated model in a procedural programming language. As well, this environment readily allows new model features, different from those first envisioned for the model design, to be included at any stage of development. Particular simulation actions, such as the execution of a particular link on the current database or the execution of all links associated with one node, can be triggered through the rule interpreter. This allows incremental testing of a model during the design process.

Features

36. When creating a model the user interface with layered menus allows for the easy creation, deletion, or alteration of links and nodes. A new attribute for individuals can be added to an existing model, which will then show up in menus throughout the model building utility as well as in structures already built. A universal deletion of an attribute is also supported, but the user would have to modify the remaining rule bases and structure information to ensure model consistency and integrity. In addition, the utility supports the cloning of objects for streamlining model development.

Output

Reporters and Probes

37. GeM provides flexible reporting mechanisms. One mechanism is called a PROBE, through which GeM can record counter values associated with either links or nodes, and record them to a file. Another mechanism provided is called a REPORTER, basically a database query. Reporters can be designed to match on individuals within a time slice. Counts or tallies of these matches, over time slices, are typically saved to a file. One reporter, for instance,

could be used to create age histograms for each time slice, of individuals:

a. of sergeant rank;
b. who have more than 2 years time in engagement; and,
c. who have between 3 and 5 years time in qualification level.

38. The output files of probes and reporters can be changed into Lotus compatible spreadsheet files for further analysis or output as graphs, by a utility provided with the model. The reporting mechanisms can be as detailed as desired, and the flexibility allows for the reporting to occur at any point within a time slice.

Logdbs

39. GeM allows for a complete dump of the model including individuals at the end of any time slice during a simulation. These dumps or logdbs can subsequently be used as a starting point for other runs examining different options. A benefit of the logdbs pertains to the use of a built-in database query facility which permits the user to examine the logdbs with custom queries that correspond to a reporter. This query facility is immensely useful whenever additional output or follow up analysis is required.

Running a Model

40. Once the model information is saved, including run parameters which control both the number of time steps and the creation of detailed logs, an execution utility is called to run the entire simulation.

41. The utility runs on a fast 80386 IBM compatible, using PDC Prolog in the OS/2 operating system. In one scenario it requires about thirty minutes to model a group of 350 individuals over a ten year time period. Although this is not breakneck speed, it is respectable, and the model provides the immense measure of flexibility required for analysis.

Application Example

42. GeM has features not found in comparable modelling tools, and it is not obvious to the first-time user how to proceed with model building. A straightforward application taken from a military career progression scenario is used to show how a basic model might look, using GeM. The graphical representation of the model is presented in the following Figures, where model links are represented by arrows, and nodes are represented by rectangles.

43. Figure 8 captures graphically the career progression within the military where promotions above the rank of Corporal are "pull" driven while promotions up to that rank are "push" driven. The Figure also portrays voluntary attrition tailored to rank characteristics. Career progression is modelled conceptually by associating nodes with the rank, and links with either the promotion process itself or with the voluntary attrition. However it should be noted that one node and link combination forms a separate but related process disconnected from all the rest. This combination handles the "push" promotion to Cpl and can be viewed as an internal process within the PTE & CPL node (27).

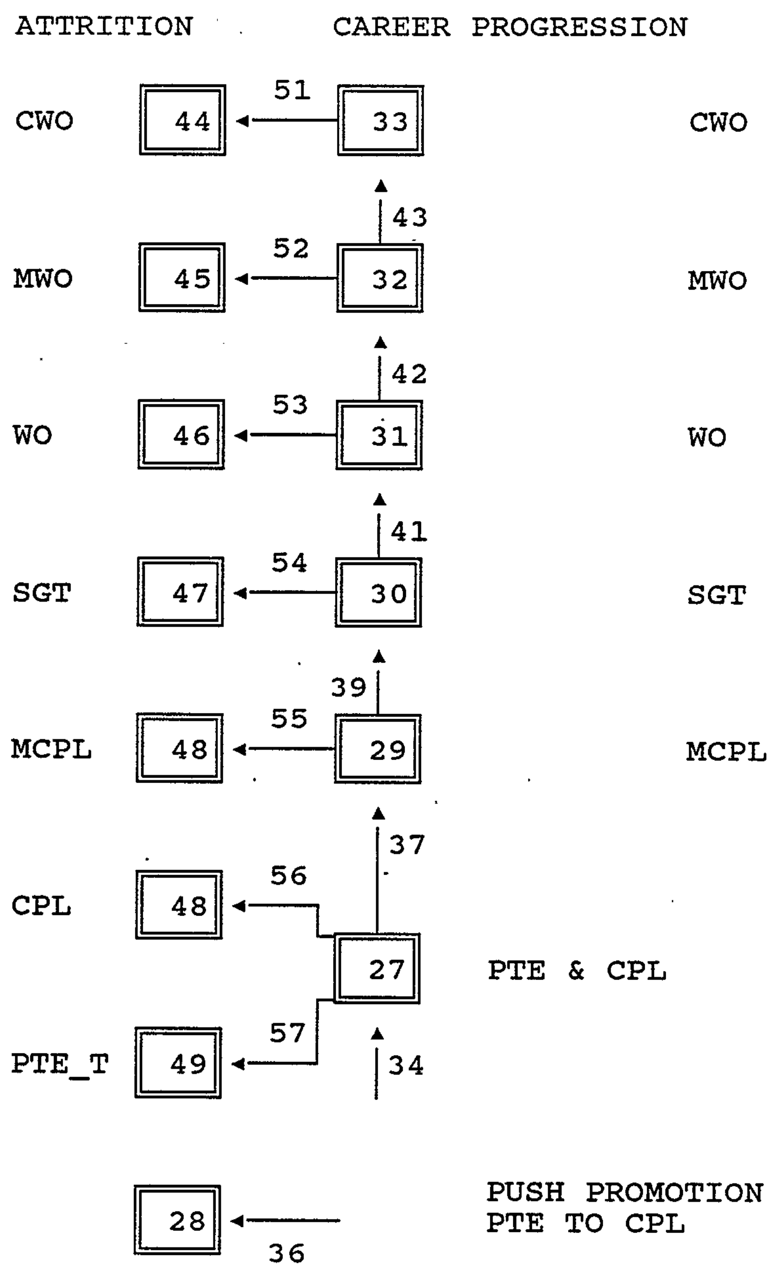

Figure 8 - Overlay Set One.[1]

[1]All numbers in the figure correspond to the object identifier within the model and are used for cross-referencing.

44. Figure 9 portrays the engagement conversion processes under the Other Ranks Career Development Plan (ORCDP), the involuntary release processes, and the aging process. Once again these different processes are modelled conceptually. In the ORCDP engagement conversion process, nodes represent the various engagements (BE, IE, IPS, CE and EXT); and links represent transitions from one engagement to another. As this process is independent of career progression it can be represented as a separate overlay; simplifying the modelling effort by reducing the number of complications in relating the different processes. The user must still ensure integrity between the various overlays, however. For example, if the engagement overlay results in the release of a sergeant, that release must be accommodated in the sgt node of the overlay handling career progression; typically achieved by sending a message to the appropriate node in the affected overlay as the release occurs.

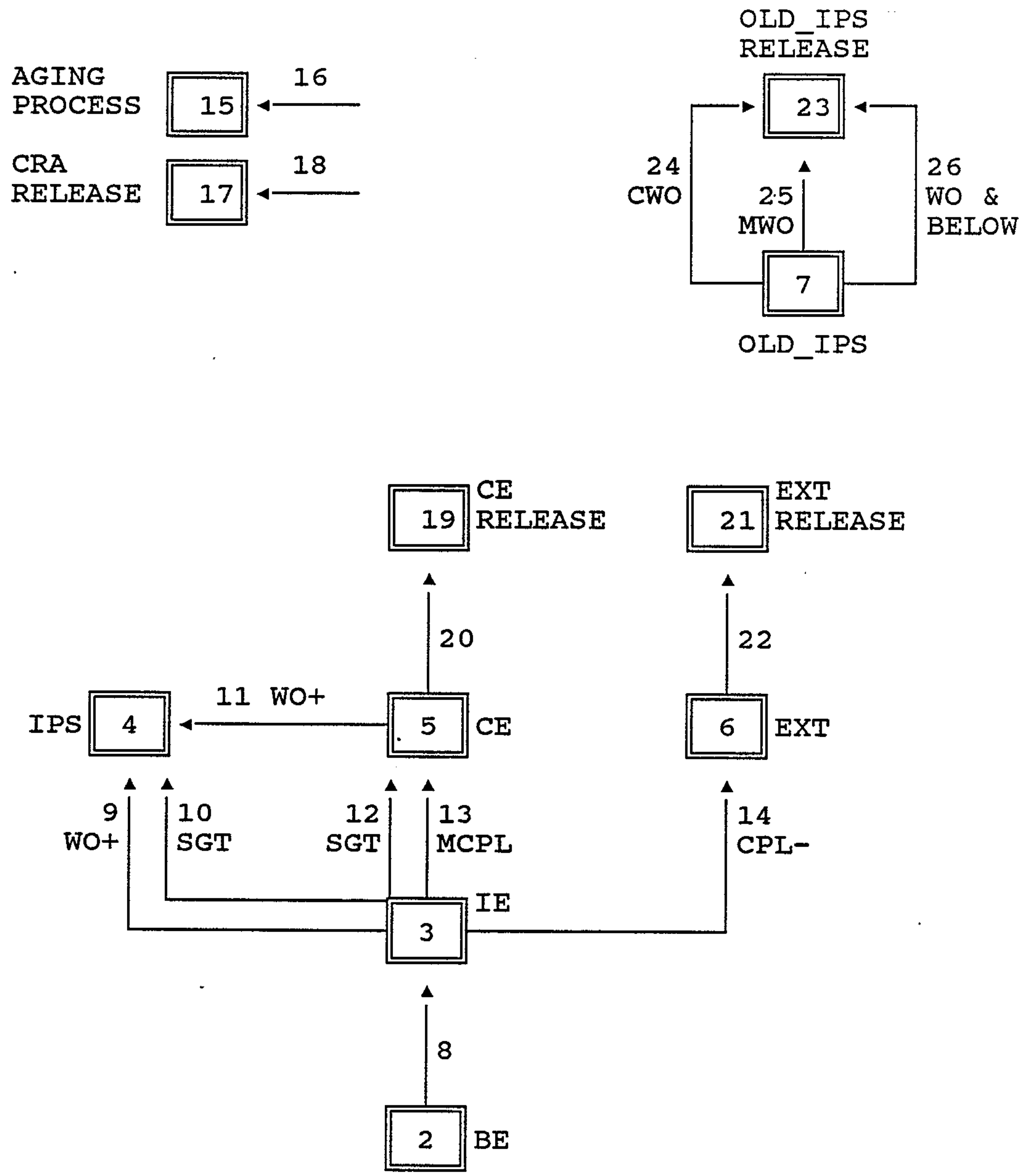

Figure 9 - Overlay Set Two.[2]

[2]All numbers in the figure correspond to the object identifier within the model and are used for cross-referencing.

CONCLUSION

45. GeM has now been used for almost a year, and user experience has been most positive. GeM responds to the needs for which it was designed. The vastly increased flexibility and capability provided by the modelling utility has not only met, but has exceeded expectations of its capacities.

46. Major model modifications which would have taken months to incorporate into D Man A's previous models can be undertaken in days with GeM. Structural changes can often be completed within hours, and parameter value changes completed within minutes. This ease of use allows the design of models far more complex than was previously possible.

47. The cost associated with the inherent power and flexibility of GeM is its steep learning curve. However, with time, a user more fully understands its broad capabilities. Once mastered, GeM is simple to use.

48. The GeM environment is easy to maintain and, because of the nature of its design, naturally lends itself to enhancements both for flexibility (more generic) and efficiency. The success of GeM has inspired further ideas for improvements. Through enhancements the potential and scope of GeM are being continually expanded. These enhancements are easily incorporated into GeM as a result of its clean design and the object-oriented nature of the utility. Although GeM was designed primarily for Human Resource Management modelling, it holds promise as a useful base for a far wider range of simulation models, both within and outside of the Human Resource Management realm.

#508439