Appendix I

INTERPRETIVE STRUCTURAL MODEL FOR THE CASE STUDY: ANIMAL TAXONOMY

A hierarchy is a system of organization in which elements form a tree-like structure.

The element set S is

1. Penguin
2. Albatross
3. Ungulate
4. Non-flying Birds
5. Flying Birds
6. Ostrich
7. Bird
8. Carnivore
9. Animal
10. Zebra
11. Cheetah
12. Giraffe
13. Tiger
14. Mammal

Using the contextual relation "is a" leads to a structural self-interaction matrix which can be refined using Interpretive Structural Modelling to structure the hierarchical knowledge as depicted in Fig. AI.1.
FIG. A1.1 INTERPRETIVE STRUCTURAL MODEL FOR THE CASE STUDY: "ANIMAL TAXONOMY"
Appendix II

STRUCTURAL CONCEPTS AND GRAPH THEORY [SAG77]

Nets

The theory of nets is based on the following four primitives:

1) A set \( P \) of elements called points \( p \).
2) A set \( R \) of elements called directed lines \( r \).
3) A function \( f \) whose domain is \( R \) and range is in \( P \).
4) A function \( s \) whose domain is \( R \) and range is in \( P \).

A simple net is shown in Fig. AII.1.

Axioms

The following two axioms are assumed while developing the theory of nets such that definition of a net is meaningful:

1) \( P \) is finite and non empty.
2) \( R \) is finite.

Loop

A line \( r_i \) is called a loop if the first and the second points of the line are the same. Mathematically this is written as
FIG. AII.1 A SIMPLE NET

FIG. AII.2 A NET DEPICTING LOOPS AND PARALLEL LINES
Parallel lines

Two lines are said to be parallel if they have the same first and second points, i.e.,

\[ fr_i - fr_j \]
\[ sr_i - sr_j \] \hspace{1cm} \ldots (AII.2)

Example 1

In Fig. AII.2:
- \( r_5 \) & \( r_6 \) are both loops,
- \( r_5 \) & \( r_6 \) are parallel to each other,
- \( r_1 \) is parallel to \( r_3 \),
- \( r_2 \) is not parallel either to \( r_1 \) or \( r_3 \).

Relations

A net with no parallel lines is defined as a relation.

Notation

If the given relation holds for two elements then it is represented as

\[ p_i R p_j \] \hspace{1cm} \ldots (AII.3)
else it is represented as
\[ p_i \bar{R} p_j \] \hspace{1cm} (AII.4)

Properties:

A relation \( R \) is \textit{reflexive} if every point \( p_i \) is on a loop:
\[ p_i \ R \ p_i \ \forall p_i \in P \] \hspace{1cm} (AII.5)

A relation \( R \) is \textit{irreflexive} if no point \( p_i \) is on a loop:
\[ p_i \bar{R} p_i \ \forall p_i \in P \] \hspace{1cm} (AII.6)

A relation \( R \) is \textit{symmetric} if
\[ p_i \ R \ p_j \rightarrow p_j \ R \ p_i \ \forall p_i, p_j \in P. \] \hspace{1cm} (AII.7)

A relation \( R \) is \textit{asymmetric} if
\[ p_i \ R \ p_j \rightarrow p_j \bar{R} p_i \ \forall p_i, p_j \in P. \] \hspace{1cm} (AII.8)

A relation \( R \) is \textit{transitive} if
\[ (p_i \ R \ p_j \land p_j \ R \ p_k) \rightarrow p_i \ R \ p_k \ \forall p_i, p_j, p_k \in P \] \hspace{1cm} (AII.9)

A relation \( R \) is \textit{complete} if for every \( p_i, p_j \) either
Digraphs

A digraph is an irreflexive relation. The theory of digraphs is based on the following four axioms:

1) The set of points $P$ is finite and non-empty.
2) The set of lines $R$ is finite.
3) No two distinct lines are parallel.
4) There are no loops.

Example 2

Fig. AII.3 shows a transitive, asymmetric, complete digraph.
Fig. AII.4 shows an intransitive digraph.
Fig. AII.5 shows a transitive digraph.
Fig. AII.6 shows a minimum line digraph corresponding to Fig. AII.5.
FIG. A II.3 TRANSITIVE, ASYMMETRIC, COMPLETE DIGRAPH

FIG. A II.4 INTRANSITIVE DIGRAPH

FIG. A II.5 TRANSITIVE DIGRAPH

FIG. A II.6 MINIMUM LINE DIGRAPH CORRESPONDING TO FIG. A II.5
Matrix Representation of Digraphs

Consider two points \( p_i \) and \( p_j \). We can conclude that either

\[ p_i R p_j \lor p_i \overline{R} p_j \]  

\[(\text{AII.11)}\]

This suggests a binary representation. Hence we can represent a digraph by a binary matrix \( A \), called the adjacency matrix. The matrix representation of the digraph of Fig. AII.6 is shown below:

\[
A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

\[(\text{AII.12)}\]

Reachability

If there is a path from \( p_i \) to \( p_j \) then we say that \( p_j \) is reachable from \( p_i \). The number of lines in the path \( p_i \) to \( p_j \) is called the length of the path.

In Fig. AII.6, \( p_4 \) is reachable from \( p_3 \) by a path of length 1, from \( p_2 \) by a path of length 2 and from \( p_1 \) by a path of length 3.

Reachability is an intuitive concept if the relation being considered is transitive:

\[\text{Every point is reachable from itself by a path of length 0.}\]
Hence, to obtain a matrix which represents reachability of all paths of length 0 and 1 we must add the identity matrix, I, to A.

\[
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0
\end{bmatrix}
\]

\[
A + I = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1
\end{bmatrix}
\] ....(AII.13)

Now multiplication of this matrix with itself (where all the operations are boolean) will result in

\[
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0
\end{bmatrix}

(A + I)^2 = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\] ....(AII.14)

Equation AII.14 describes the reachability for all paths of length two or less.

Raising \((A+I)^2\) to next power gives

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0
\end{bmatrix}

(A + I)^3 = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}
\] ....(AII.15)

Multiplying this further by \((A+I)\) will give the same matrix, i.e.,

\[(289)\]
\[
(A + I)^4 = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 \\
\end{bmatrix} \quad \ldots \text{(AII.16)}
\]

Hence we observe that
\[
(A + I)^3 - (A + I)^4 - (A + I)^5 - \ldots \text{ etc.} \quad \ldots \text{(AII.17)}
\]

This shows that, for the digraph in Fig. AII.6, the path with maximum length is 3. In general terms
\[
(A + I)^{r-2} * (A + I)^{r-1} - (A + I)^r - P \quad \ldots \text{(AII.18)}
\]

where P is called the transitive closure of A.

**Strongly Connected Digraph**

If a digraph or a subset of a digraph has a universal matrix for its reachability matrix that digraph or subset of digraph is said to be strongly connected.

**Observation**

1) A unique reachability matrix corresponds to a given adjacency matrix.
2) There is no unique adjacency matrix corresponding to a given reachability matrix.
Appendix III

THE INTERPRETIVE STRUCTURAL MODELLER

The Interpretive Structural Modeller is a user friendly knowledge-based multi-purpose software package, the important features of which are discussed below. Fig. AIII.1 depicts the architecture of the developed software.

The Interpretive Structural Modeller has been developed using both Turbo-C and Turbo-Prolog. The OOA shell has been developed in Turbo-Prolog and the Interpretive Structural Modeller in Turbo-C.

Software scopes can be created/edited using the software scope editor. Subsequently, an object oriented analyzer can be used to select relevant objects from the software scope. The basic principle is that the software scope is analyzed, and unwanted words are filtered with the help of a knowledge database. Duplicated words are then eliminated and the nouns are selected. Further, the user is prompted to pick up the relevant objects in the problem/solution space through an interrogative process. The selected objects are then displayed. Fig. AIII.2 shows a screen print of the object-oriented analyzer during a typical session.

ISM may then be automatically invoked from the main menu and C-routines generate a minimum edge digraph representation of the objects in question. The Interpretive Structural Modeller is an independent system and works in the following sequence:
Menu based User Interface

Software Scope Editor

User Refinement

Object Oriented Analyzer

Knowledge Database

Refined Object Datafile

Interpretive Structural Modeller

Graphics Engine

Graphic ISM Images

FIG. III.1 ARCHITECTURE OF INTERPRETIVE STRUCTURAL MODELLER.
Interpretive Structural Modeller

<table>
<thead>
<tr>
<th>Editing Facilities</th>
<th>OO Analysis</th>
<th>Utilities</th>
<th>ISM</th>
</tr>
</thead>
</table>

Analyze the software scope

Analyzed scope text

A central computer transmits NC_blocks to NC_software which reads each NC_block and stores it in an NC_program_file. NC_blocks are read from the NC_program_file and decomposed into control_words for position and special control functions.

Selected Objects

nc_blocks  nc_software  nc_program_file  control_words
Number of objects selected >> 4
Press any key to return to Main Menu...

Esc: Quit the system

Fig.AIII.2  Object-oriented Analyzer Screen Print
1. Prompts for the number of objects.
2. Asks for the basic contextual relationship involved.
3. Asks the names of the objects
4. Generates an automatically labelled and sized grid to facilitate the gathering of data on the interrelationships. In this phase the top of the screen displays the basic contextual relationship and the two objects that are involved in the current query.

Program computations are of two basic kinds:


2. Graphical: From the minimum edge adjacency matrix generation, the output is used to generate the screen display.

Figs. AIII.3, and AIII.4 show screen prints of the SSIM interrogation process and the resulting ISM respectively.
Structured Self Interaction MAtrix Data

Fundamental relationship: depends_upon

controlwds commands

NCsoftware
NCblocks
NCprogfile
controlwds

commands

Fig.AIII.3  SSIM Interrogation Screen Print
Fig.AIII.4 Exemplar ISM Graphic Screen Print
Appendix IV

THE SOFTWARE PROJECT PLANNER

The Software Project Planner is a neural network based system dynamics simulation package which provides versatile support to the project manager in planning and control of the development of software systems. Fig. AIV.1 shows the general architecture of the developed system.

With the help of a user interface, the user is expected to provide data on the labor rate and productivity index of the organization. Subsequently, neural network support may be invoked to provide an estimate of the required manpower buildup index which is a function of the task concurrency, schedule pressures and the application complexity. Smallest, most likely, and largest SLOC estimates from expert delphi polls are provided to compute the minimum time solution.

The management decision support neural network, ANN, took 22208 repeated pattern presentations in 347 epochs to be trained\(^5\) to an error tolerance of 0.057679. The decision support system, DSS, conducts checks on the computed values with the help of user defined ceilings on the project. Trade-off opportunities are indicated and advice on how to possibly fit the project within the ceilings provided.

\(^5\) Training of the neural network was carried out on a PC-AT 386 computer and took 8643 seconds (2:24:03 hrs.)
FIG. AIV.1 ARCHITECTURE OF THE SOFTWARE PROJECT PLANNER
The user may then invoke the graphics engine to generate check-plots of the projected graphs for the development of source code, the development effort, the life-cycle effort and the manpower profile for the selected planning point as shown in Fig. AIV.6.

MBI selection support is provided by the neural network, ANN, with "fuzzy" inputs as shown in the computer screen print of Fig. AIV.2. Figures AIV.2 through AIV.6 show a typical run of the simulation software on the computer for a software project with specifications as in Fig.AIV.2. Fig.AIV.3 shows the four ceilings captured by the software prior to the minimum time solution of Fig.AIV.4. Fig.AIV.5 depicts the support provided to the management by the decision support system DSS. At this point, the management can override the decision support system and proceed with the simulation as shown in Fig. AIV.6.
Project Description

Title: Office Automation

Labor Rate (Rs/Yr): 50000
Primary Language: Pascal

Productivity Index: 12
Productivity Parameter: 10946.00

Manpower Buildup Index: Suggest from Neural Net? (y/n) y

Task Concurrency: (Low/Low-Med/Med-Hi/Hi) - (1-4): 2
Application Complexity: (Low/Low-Med/Med-Hi/Hi) - (1-4): 3
Schedule Pressure: (Low/Low-Med/Med-Hi/Hi) - (1-4): 2

MBI Selected: 3
Manpower Buildup Parameter: 26.90

Override? (y/n): n

Size Estimate:
Smallest: 20000
Most Likely: 30000
Largest: 40000
Expected: 30000
Standard Deviation: 3333

Special skills parameter: 0.280000

Fig.AIV.2 Project Description Screen Print

Please enter the following respective ceilings for the Project

Maximum Development Cost (x 1000 Rs.): 65
User Delivery deadline (x Months): 12
Maximum available manpower (People): 7
Maximum Allowable risk (%): 5

Fig.AIV.3 Project Ceilings Interrogation Screen Print
Minimum Time Solution
Project > Office_Automation Thu Sep 05 18:17:04 1991

<table>
<thead>
<tr>
<th>MANAGEMENT METRIC</th>
<th>EXPECTED VALUE</th>
<th>STD DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM SIZE (SOURCE STATEMENTS)</td>
<td>30000</td>
<td>3333</td>
</tr>
<tr>
<td>MINIMUM DEVELOPMENT TIME (MONTHS)</td>
<td>11.550152</td>
<td></td>
</tr>
<tr>
<td>DEVELOPMENT EFFORT (MM)</td>
<td>47.973524</td>
<td></td>
</tr>
<tr>
<td>LIFECYCLE EFFORT (MM)</td>
<td>287.841145</td>
<td></td>
</tr>
<tr>
<td>DEVELOPMENT COST (x 1000 Rs)</td>
<td>1199.34</td>
<td></td>
</tr>
<tr>
<td>PEAK MANPOWER (People)</td>
<td>15.115341</td>
<td></td>
</tr>
<tr>
<td>PRODUCTIVITY (SLOC/MM)</td>
<td>3206.897064</td>
<td></td>
</tr>
<tr>
<td>DIFFICULTY (MY/Y^2)</td>
<td>25.891590</td>
<td></td>
</tr>
<tr>
<td>RISK (PERCENT)</td>
<td>8.020939</td>
<td></td>
</tr>
</tbody>
</table>

Press any key to continue.

Fig.AIV.4 Minimum Time Solution Screen Print

Please Wait! Conducting Ceilings Check ...

WARNING! Ceilings Violation!!

Decision Support System Suggestion/Explanation
Cost, Peak manning, Risk has(ve) exceeded ceiling(s)!

Conflict situation: decreasing td increases Mp and vice-versa
Conflict situation: decreasing Mp increases the risk and vice-versa
Conflict situation: if the cost is reduced by decreasing TC, td increases, thereby increasing the risk

Modify parameters? (y/n) >>

Fig.AIV.5 Decision Support System Screen Print

(301)
Generic Lifecycle Effort & Development Subcycle Effort (MM)

Cumulative Source Code Produced (x1000)

Fig. AIV.6 Dynamic Check Plots Screen Print
DISCUSSION ON THE STABILITY OF THE HOPFIELD NETWORK [HOP84]

Consider the "computational" energy function:

$$ E = -\frac{1}{2} \sum_i \sum_j T_{ij} v_i v_j - \sum_i I_i v_i + \sum_i \left( \frac{1}{R_i} \right) \int_0^{v_i} g_i^{-1}(v) \, dv \quad \ldots \quad (AV.1) $$

s.t.

$$ \frac{dE}{dv_i} = - \left( \sum_j T_{ij} v_j + I_i - \frac{g_i^{-1}(v_i)}{R_i} \right) \quad \ldots \quad (AV.2) $$

$$ \frac{dE}{dv_i} = -C_i \frac{du_i}{dt} \quad \ldots \quad (AV.3) $$

Then

$$ \frac{dE}{dt} = \sum_i \frac{dE}{dv_i} \cdot \frac{dv_i}{dt} \quad \ldots \quad (AV.4) $$
\[- \sum_i \left( -C_i \frac{du_i}{dt} \right) \frac{dv_i}{dt} \quad \ldots (AV.5)\]

\[- \sum_i C_i \frac{d}{dt} \left( g_i^{-1}(v_i) \right) \frac{dv_i}{dt} \quad \ldots (AV.6)\]

\[\frac{dR}{dt} = -\sum_i C_i g_i^{-1}(v_i) \left( \frac{dv_i}{dt} \right)^2 \quad \ldots (AV.7)\]

and thus

\[\frac{dR}{dt} = 0 \Rightarrow \frac{dv_i}{dt} = 0 \quad \ldots (AV.8)\]

Note that this is valid only for a symmetric [T] matrix. The Hopfield network states thus evolve in such a manner so as to minimize the E function, which is specific to a particular application of the system. E is minimized when \(dE/dt = 0\), which has been mathematically shown to imply that \(dv_i/dt = 0\) in that state (\(V_i\) is the output state of the \(i^{th}\) neuron). Therefore as \(dE/dt\) approaches zero the voltage states of the network approach a point which is a stable state of the network.
Appendix VI

THE HOPFIELD SIMULATOR

The Hopfield simulator is a software package developed specially for the purpose of studying the dynamic trajectories of state and sensitivity state vectors of the Hopfield network, and for the minimal sensitivity design of this class of recurrent networks.

Fig. AVI.1 shows the architecture of the Hopfield simulator using which, simulation studies may be conducted on the state voltage and state sensitivities of the network. Second order sensitivities are employed to conduct a gradient descent as discussed in Chapter 5 to yield a minimal sensitivity design of the network. A typical network configuration would be stored in the format shown in Table AVI.1.

Fig. AVI.2 shows an output screen print of the software during a typical simulation run. Default simulation parameters may be changed and simulation speed slowed down to observe parameter changes more carefully.

| 2         | <- number of neurons |
| 1.0 1.0   | <- amplifier input resistance |
| 0.01 0.01 | <- amplifier input capacitance |
| 1.0 -1.0  | <- external input excitation currents |
| 0.0 -1.0  | <- Tij connection matrix |
| -1.0 0.0  |

Table AVI.1  Network file format
FIG. VI.1 ARCHITECTURE OF HOPFIELD SIMULATOR
Hopfield Neural Net Simulator - Results Screen
Thu Sep 05 19:03:11 1991.

<table>
<thead>
<tr>
<th>Epoch#</th>
<th>Neuron count</th>
<th>Time step</th>
<th>Max tau</th>
<th>Simulation time</th>
<th>Scale factor</th>
<th>Elapsed time</th>
<th>Lambda</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>2</td>
<td>0.0000100</td>
<td>0.0000200</td>
<td>0.000700</td>
<td>2.00</td>
<td>6.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

Neuronal Transfer Characteristic - Sigmoidal

<table>
<thead>
<tr>
<th>Neuron #</th>
<th>Input Voltage</th>
<th>Output Voltage</th>
<th>Sensitivity (w.r.t. r1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.400839428</td>
<td>+0.401110675</td>
<td>-6.462640e-08</td>
</tr>
<tr>
<td>1</td>
<td>-0.401102653</td>
<td>+0.401047445</td>
<td>-6.469462e-08</td>
</tr>
</tbody>
</table>

Gradient Descent

Iteration Number → 1  
Error function: 0.012066

Delta r/C/Lambda/T = -0.000000 -0.000000 0.099990 0.001422
r/C/lambda/T = 100.000000 0.010000 1.099990 -499.998578

Fig.AVI.2 Dynamic Simulation and Gradient Descent Results Screen Print