# Concepts, Tools and Applications for Design reasoning Recording- The AAITP Detract

Karl Reed[1] and Torab Torabi[2,3]

[1]Gastwissenshaftler, Fraunhofer-Gesellschaft, IESE (August 2002-May 2003

[2]La Trobe University, Department of Computer Science and Computer Engineering

[3]The work of Jacob Cybulski, David Cleary, Jamie Lenehan, Jason Baragry and Fred Brikich is also involved

12/2/2006 rev 4

# This Presentations Goals….

**Examine Means of Assisting Architecture recovery by...**

**Using Design Reasoning Recording as a possible aid**

A. Describe the Problem,

B. Introduce Design Reasoning Recording, DETRACT

C. Introduce the AAITP Compilable Restricted natural Language (CRNLP) approach, (Controlled Languages)

D. Present the AAITP HyperCASE framework,

E. Suggest some lines of investigation and possible tool development,

*BUT! Not necessarily in that order!*

*("Real Programmers write programs, not documentation. Leave that to the maintenance people" from Ed Post, Real Programmers Don't Use Pascal", Datamation V.29 N. 7 July 1983 pp263-265)*

*(References are listed at the end…)*

# A. Describe the Problem(cont'd),

***§ Maintainers need to understand the code (and the specification, and the application domain)***

*§ IF some defined process was followed, that included documentation, this may help--*

*§IF the documentation aids an understanding of WHY the code is like it is, this may help*

***§ Recovering the architecture from a system is important because…***

*§ It may help with product-line development, assisting the process of salvaging code,*

*§ It may help with modifying code in the maintenance process*

***So, it would help if the a' priori architecture used at the beginning of the design process could be found in the code…***

***BUT IT OFTEN DOESN'T HAPPEN***

- ☞ *Documentation is often completed AFTER the system is finished, if at all*
- ☞ *Documentation usually shows WHAT has been done, and not WHY?*
- ☞ *Re-engineering researchers consistently remark that the product architecture does not relate to the a priori architecture.*

***Maintainers must "comprehend" the program, and re-construct the reasons for the implementation***

# A. Describe the Problem(cont'd),

## § *What should be recorded, and how?*

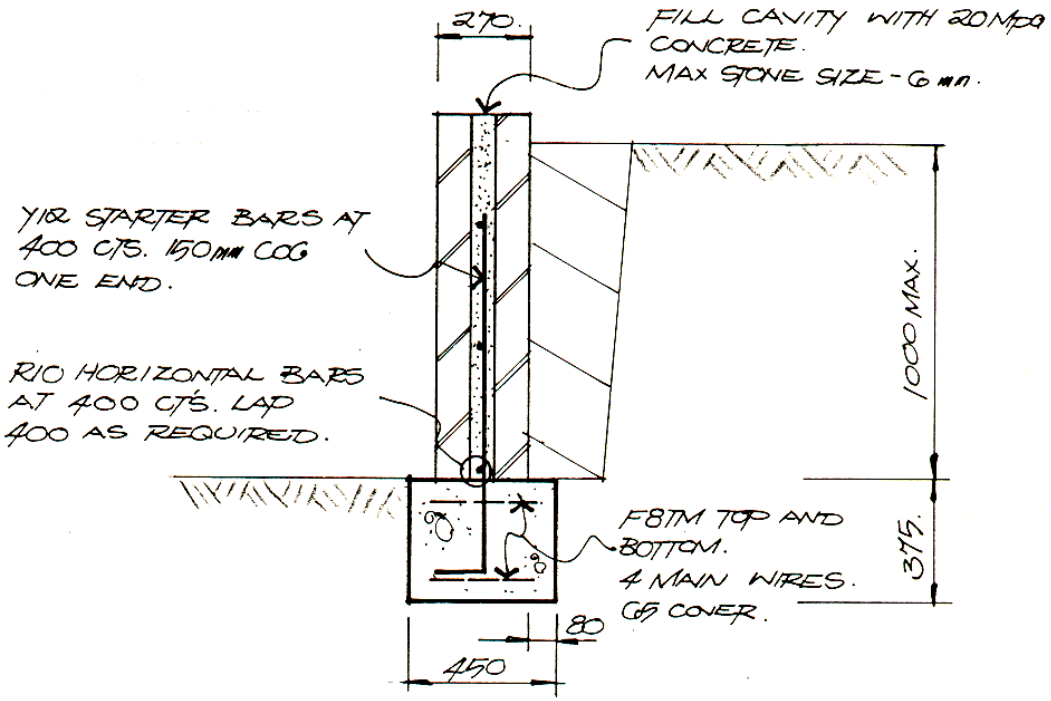
*¶ This a hard question…*

*The real answer is "..those things that designers do when making decisions…."*

*-- Research on this is not very conclusive.. Something IESE can help with?*

## *A MINOR DIGRESSION.. WHAT DO ENGINEERS DO?*

# *The result of an engineering design*




## RETAINING WALL DETAIL.

NOTES.

PROVIDE MOISTURE BARRIER AND DRAINAGE BEHIND WALL TO COUNCIL REQUIREMENTS.

USE 18 1/4 : 3 MORTAR FOR WALL CONSTRUCTION.

# *Engineering is..*

*-->> "A directed process of decision making leading to the design of a realisable artefact in which criteria exist for choices which guarantee optimal outcomes according to some pre-determined criteria"*

*-->>"Design-reasoning Explicit…"*

*The design CANNOT be completed without performing explicit 'reasoning', which MUST be recorded, step by step..*

*-->>Hence, the documentation is an integral part of the process, NOT SOME EXTERNALLY MANDATED MANAGERIAL REQUIREMENT!*

- consider min design requirements to existing floor beams
  - for increased spans.

max span = 4800 continuous (2000) (front)

max loadings m-m

| | | | | |
|---|---|---|---|---|
| Roof | w = | 0.15 × 3.50 | = | 0.53 kl. |
| | w = | 0.25 × 3.50 | = | 0.87 |
| Ceiling | = | 0.10 × 3.50 | = | 0.35 |
| partition wall | = | | = | 0.50 |
| Floor | w = | 0.40 × 4.30 | = | 1.72 |
| | w = | 1.50 × 2.50 | = | 3.75 |
| | w = | 3.0 × 1.80 | = | 5.40 |

for existing beam = 0.20 kl

| | | | | |
|---|---|---|---|---|
| Roof | w = | 0.15 × 1.50 | = | 0.23 |
| | w = | 0.25 × 1.50 | = | 0.38 |
| Ceiling | = | 0.10 × 1.50 | = | 0.15 |
| | | | | 14.08 kl |

max wt = 30.69 kl.

$$EI_{req'd} = \frac{7.10(2.5 \times 3.68)(4.8)(4800)^3}{12.0} = 2.89 \times 10^{12}$$

EI reqd = $4.35 \times 10^{12}$
u.

for existing 2#-270 × 75 Oregon beam

$EI = 9100 \left( 118.10 \times 10^{4} \right) 2 = 2.149 \times 10^{12}$

resultant $EI = 2.20 \times 10^{12}$.

for m.s. Section $E = 2 \times 10^{5}$

try — 180 × 75 parallel flange channel

$EI = 2 \times 10^{5} \left( 14.10 \times 10^{4} \right) = 2.82 \times 10^{12}$

max deflection $= \dfrac{4.35 \times 10^{12}}{4.97 \times 10^{12}} \times 9.0 = 7.88$

max mt $= 165 \left( 157 \times 10^{3} \right) + 8.6 \times 1.40 \left( 855.47 \times 10^{3} \right) 2$

$= 46.50$ t

$> 30.69$ t

Adopt — existing 2#-270 × 75 Oregon beam
+ additional 180 × 75 parallel flange channel
1# × 10 bolt @ 600 mm c/c diagonally staggered top & bottom.

max reaction = 69.61 kn internal
(18.19 kn D.L).

max pad hight = 3000 mm

for 150 x 150 F7 Omega pad
A = 21610 mm²

$$\left[\begin{array}{l} f_c = \dfrac{0.85(3000)}{150} \\ = 17.0 \end{array} \quad \left( \begin{array}{l} T_{hu}\big|_{T+L} = 0.74 \\ \rho = 0.95 \end{array} \right) \right] \quad f_{cp} = 16.15$$

$$f_c' = 5.10 \times 1.40 \times 0.62 = 4.50 \text{ mpa}$$

$$f_c = \frac{69.61 \times 10^3}{21610} = 3.22 \text{ mpa}$$

Adopt - 150 x 150 F7 Omega support pad - internal

provide min - 600 x 600 x 350 mm depth cover pad.

Note: for end support - 110 mm and bearing to brickwork

- consider min design requirements to existing floor beams
  - for increased span - near

max span = 2400 — centre.
(2200 + 2000).

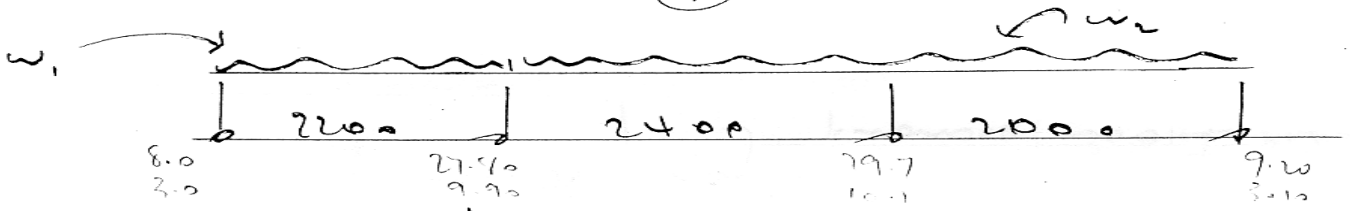
max loading m =

| | | | |
|---|---|---|---|
| Roof m = | 0.15 × 5.0 | = | 0.75 kN |
| n = | 0.25 × 5.0 | = | 1.25 |
| Ceiling = | 0.10 × 5.0 | = | 0.50 |
| part wall = | | = | 0.9 |
| Floor m = | 0.40 × 3.20 | = | 1.28 |
| n = | 1.50 × 3.20 | = | 4.80 |
| beam wt = 0.20 kN | | $w_1$ = | 9.66 kN |

3.63

| | | | |
|---|---|---|---|
| Roof m = | | = | 0.75 kN |
| n = | | = | 1.25 |
| Ceiling = | | = | 0.50 |
| part wall = | | = | 0.9 |
| Floor m = | 0.40 × 4.50 | = | 1.80 |
| n = | 1.50 × 4.50 | = | 6.75 |
| beam wt = 0.20 kN | | $w_2$ = | 12.15 kN |

4.15




max mmt = 6.00 L

$I_{req'd} = 7.14 \times 10^6 \times 2.50 = 17.85 \times 10^6 \ mm^4$

from observation

Adopt - existing 2# 270 x 75 Oregon beam on offset

provide min support

- 100 x 100 F7 Oregon post - studwall

- 110 mm end bearing to brickwall - as req'd

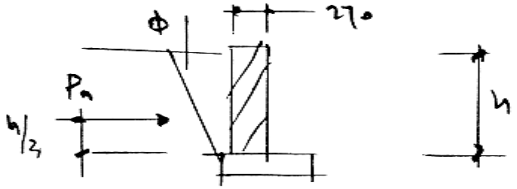
critical post to oregon post

400 x 400 x 350 mm depth core post

# RETAINING WALL.

[Note: no surcharge from existing residence applied to retaining wall]

max effective height of soil ↷ wall — 1000 —



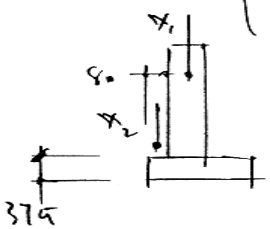

$\gamma = 19\ kN/m^3$

$\phi = 30° \longrightarrow K_a = 0.30$

$f'_c = 20\ MPa$ ↷ 28 Days

$f_{sy} = 410\ MPa \longrightarrow f_c = 210\ MPa$

loading to wall.




Wall $(x_1)$ = 8.36 + 1.56 = 9.92 kN

Soil $(x_2)$ = 18.0 × 0.08 × 1.0 = 1.44

```
RETAINING
WALL:

Pa=2.9700
MOMENT=0.9900
AST=38.8007
FOOTING WIDTH=0.4500
SAFETY FACTOR=1.8219
e=0.1128
Y=0.1122

BEARING=91.5575
```

Adopt —

270 mm cavity brickwall

Y12 bars ↷ 400 mm c/c × 900 lap vert

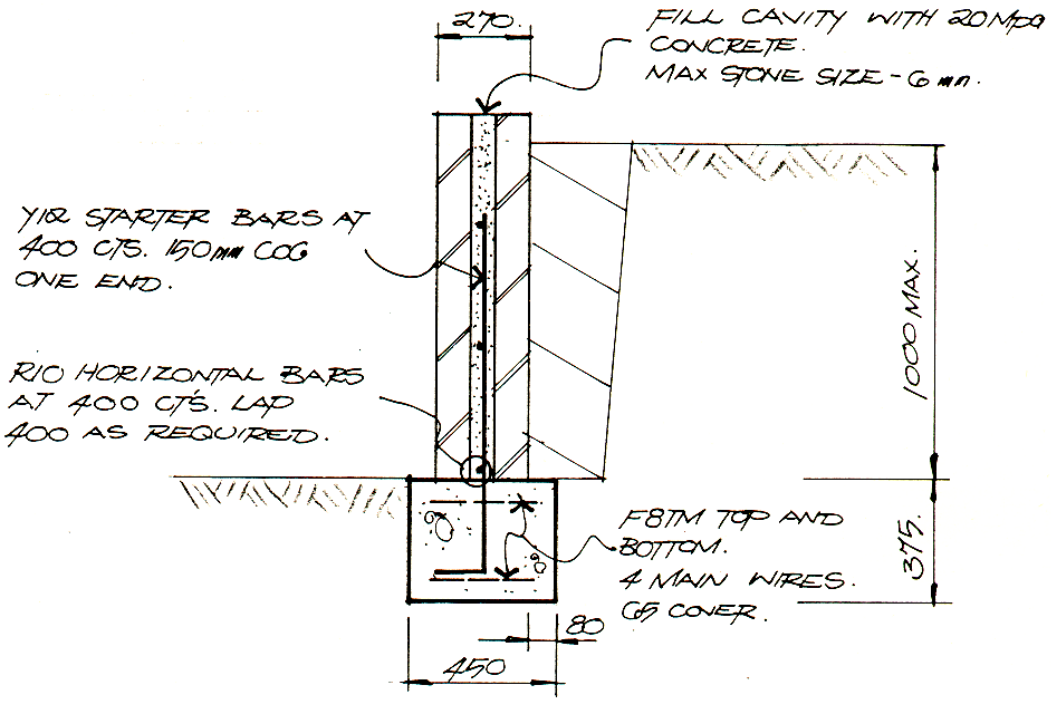
+ R10 bars ↷ 400 mm c/c horiz

provide min 450 mm × 375 mm depth cons strip footing

& Layer F8 bars top & bottom 4th maximum.

J. W. Scott & Associates Pty. Ltd.

# *The result of an engineering design*




## RETAINING WALL DETAIL.

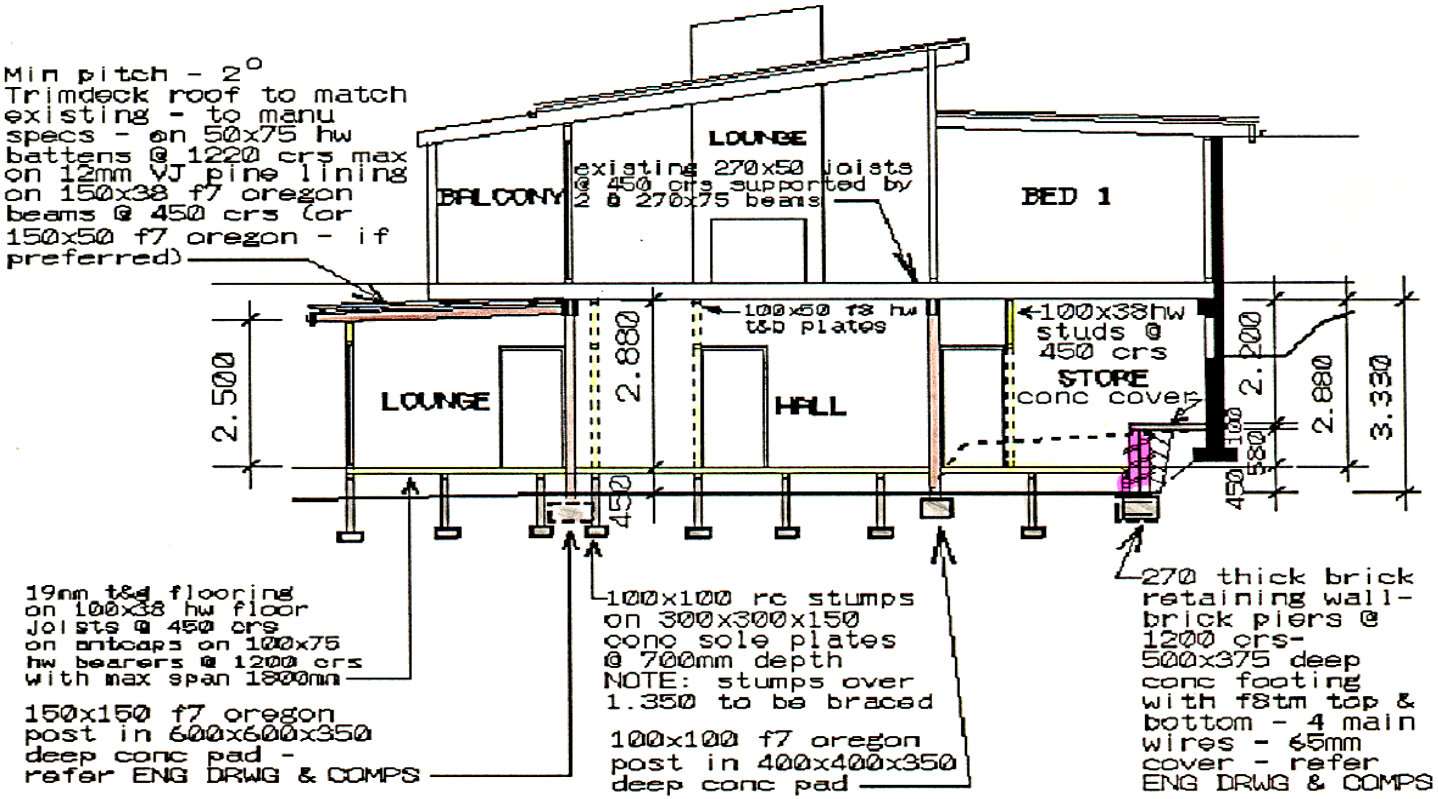
NOTES.

PROVIDE MOISTURE BARRIER AND DRAINAGE BEHIND WALL TO COUNCIL REQUIREMENTS.

USE 18 1/4 : 3 MORTAR FOR WALL CONSTRUCTION.

# SECTION A-A

NOTE: ACCORDING TO DIAMOND VALLEY COUNCIL - SOIL IS ASSUMED - STABLE-INTERMEDIATE




| Draftcorp<br>P O Box 424, Greensborough, 3088<br>tel: (03) 435 2203 fax: (03) 434 1670 | | | | PROPOSED ALTERATIONS & ADDITIONS TO EXISTING B/V HOME |
|---|---|---|---|---|
| SECTION | | | | for: MR K & MRS R REED<br>at: 150 RYAN'S RD.,<br>ELTHAM NORTH (LOT 8) |
| sheet no.: 4 | scale: 1:100 | date: 10.04.1990 | job no.: 8985 | |

## A. Describe the Problem,(cont'd)

***Maintainers must "comprehend" the program, and re-construct the reasons for the implementation***

***§ THEY RE-INVENT THE "DESIGN REASONING", which is not in the documentation-if there was any…***

***The Researcher's Solution?***

# ***Design Reasoning Recording!***

*§ Potts and Bruns (1988), Arango and Bruneau (1991), Lee (1991) Reed (1988), Reed and Torabi (1992), Ramesh and Dhar (1992), and many more later*

*All proposed that designers record the reasons for their design decisions, and proposed (and built) tools.*

# *Design Reasoning Recording!*

*(we must of course record decisions….)*

## *What are the attributes of design artefacts?*

*Human readable, possibly hand-written, hand drawn*

*Scraps of information*

*Machine-readable artefacts*

## *What kind of information do we expect collect?*

*Diagrams, Code fragments*

*Explanations… Why was this choice made, and NOT some other*

*Domain expert decisions*

## *How do we wish to use it?*

*Query the data to find…*

*Similarities.. Where a decision was made before, what is affected by a decision when it was made*

*What artefacts are related to any given decision*

# *Design Reasoning Recording!*

*What we conclude at about the "nature" of what we record..*

***It will be textual (and may be graphical)***

***It is likely to be natural langauge..*** "How many characters needed for the PIN number?"

***We need to extract "meaning" from it, so that we can query the data to find that*** "The PIN is between six and eight digits"

***Strictly speaking its hard to do this with NL….***

***Enter CRNLP!!!! And controlled langauges***

# B. DETRACT
# A Design Decision Tracking and Reasoning Tool

Torab Torabi

Jamie Lenehan

Fred Brkich

12/2/2006 rev 4

# Motivations

- Design methodologies
  - different methodologies are used
  - design may not follow a methodology
  - documentation may not reflect the design
- Design documentation
  - only describes the product not the design decisions
  - often documentation is done after software implementation
  - documentation is huge and hence difficult to use

detract

File Search Move Edit Help Options

**Artifact:** query user
**Type:** rule
**Actual:** l:/aaitp/ostar/david:user_on_q

**Change History:**
31 Oct 1996 Karl Reed (kreed)
31 Oct 1996 Abou-Torab Torabi (torab)
31 Oct 1996 Abou-Torab Torabi (torab)

**Issues** Add Del
how should priority of account be mc
how should status of user be modifie
how should type of user be verified?
how should user be defined?

**Sub Artifacts** Add Del
check user logon
logged-on
set not logged-on
User logon Status

**Goals** Add Del
count user logons

*Select artefact "Query User" All related design decision elements are displayed*

**Summary**
This rule should initialise the user logon status and then check the actual status of the user.

**Related** Define Del
[ted ](Depends To ) l:/aaitp/ostar/d:
[ted ](Depends to ) l:/aaitp/ostar/da
[detract ](relating to ) user

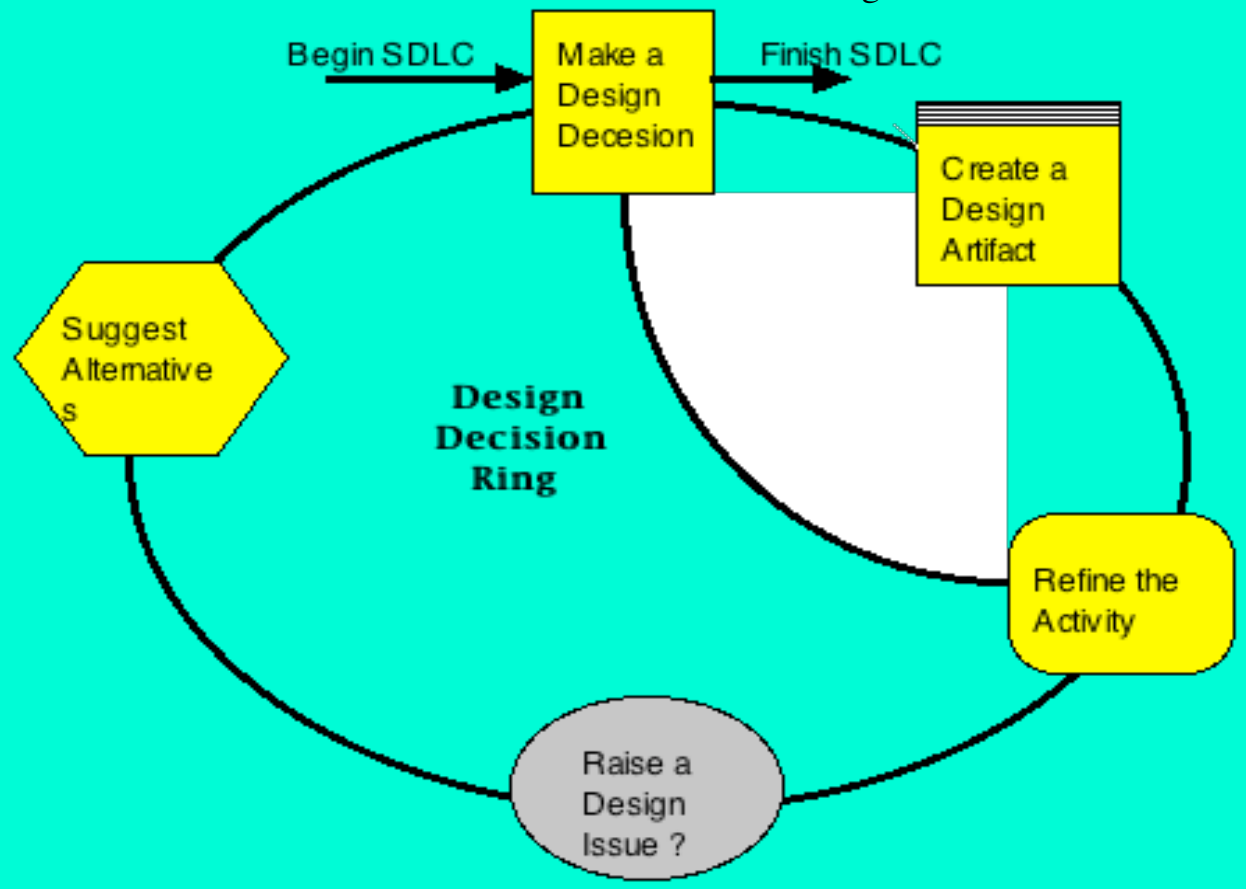
PROJECT (HyperCASE for Object Star) ARTIFACT (query user)

# Recording Design Decisions

- Time stamping
- Recording design decision information
- Maintaining history of design decisions and design artefacts
- Constructing artefact dependencies
  - design decision model
  - explicit relations
  - natural language processing

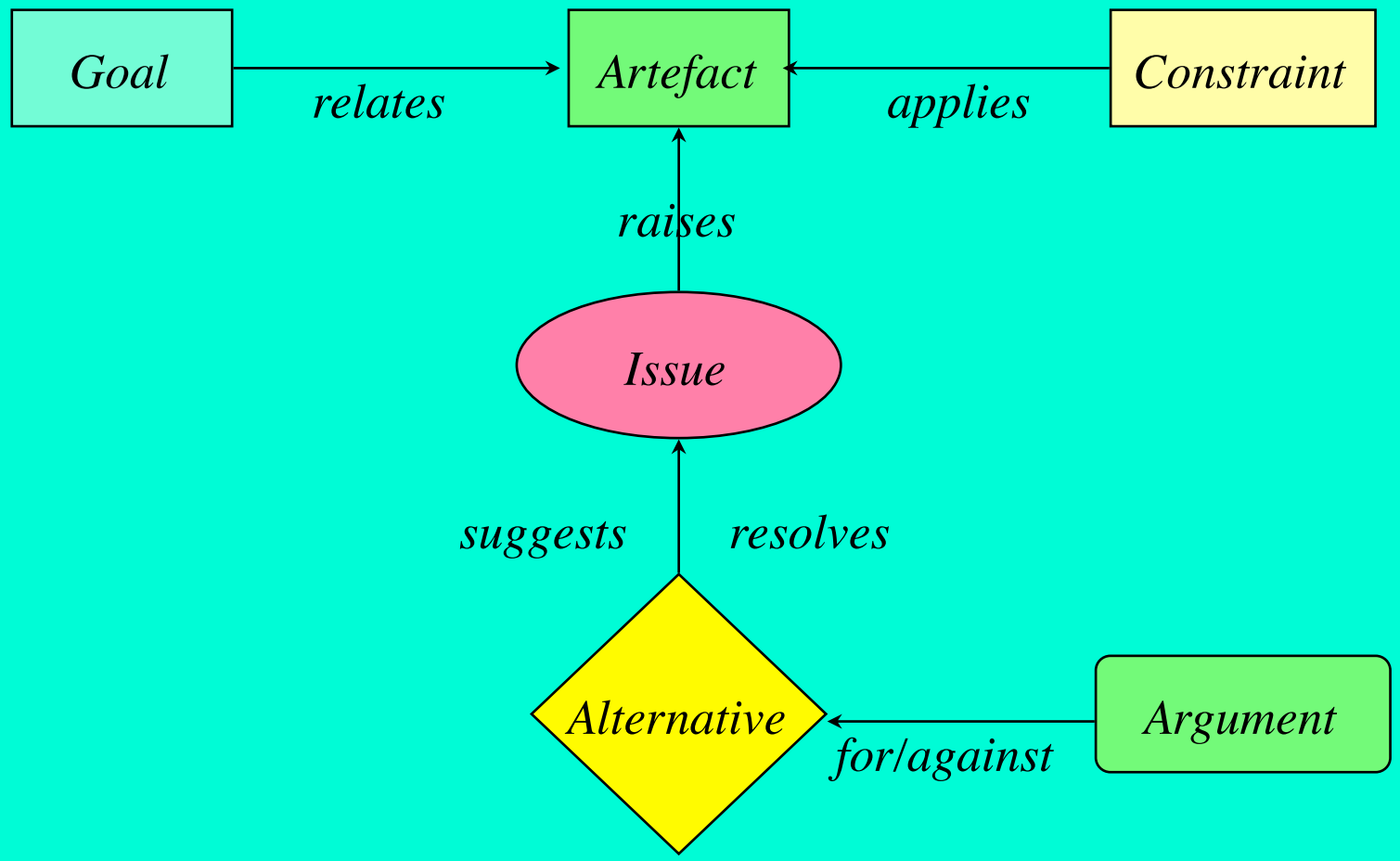
# The Design Decision Ring

Fig 5, Torab and Reed, TR028 1993




**Abstract Model for Design Decision Ennvironment and Relationship between Design Artifacts**

# Design Decision Model



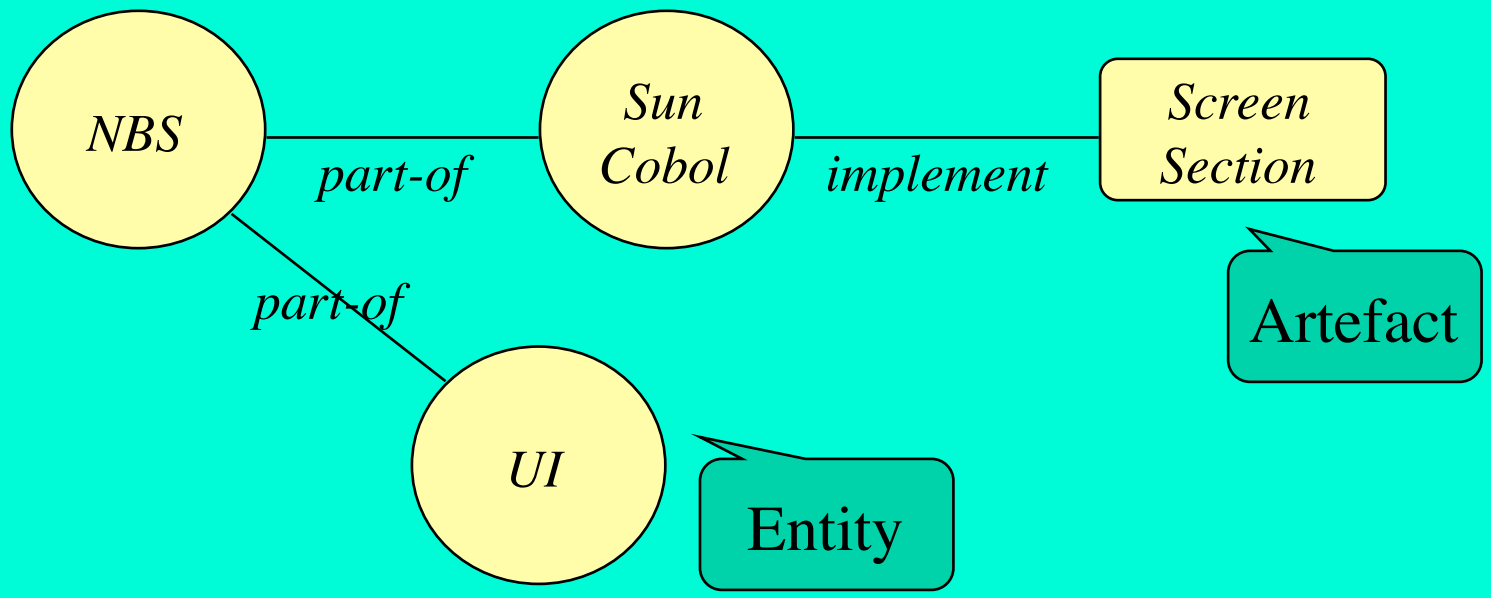
# Natural Language

- Uniquely represent design decision information
- Identify other entities in problem domain
- Identify non-explicit relations between entities
- Cross referencing of design artefacts and design decision artefacts
- Provide information for completeness and consistency checking
- Provide a basis for inferencing

# Natural Language (ctd)

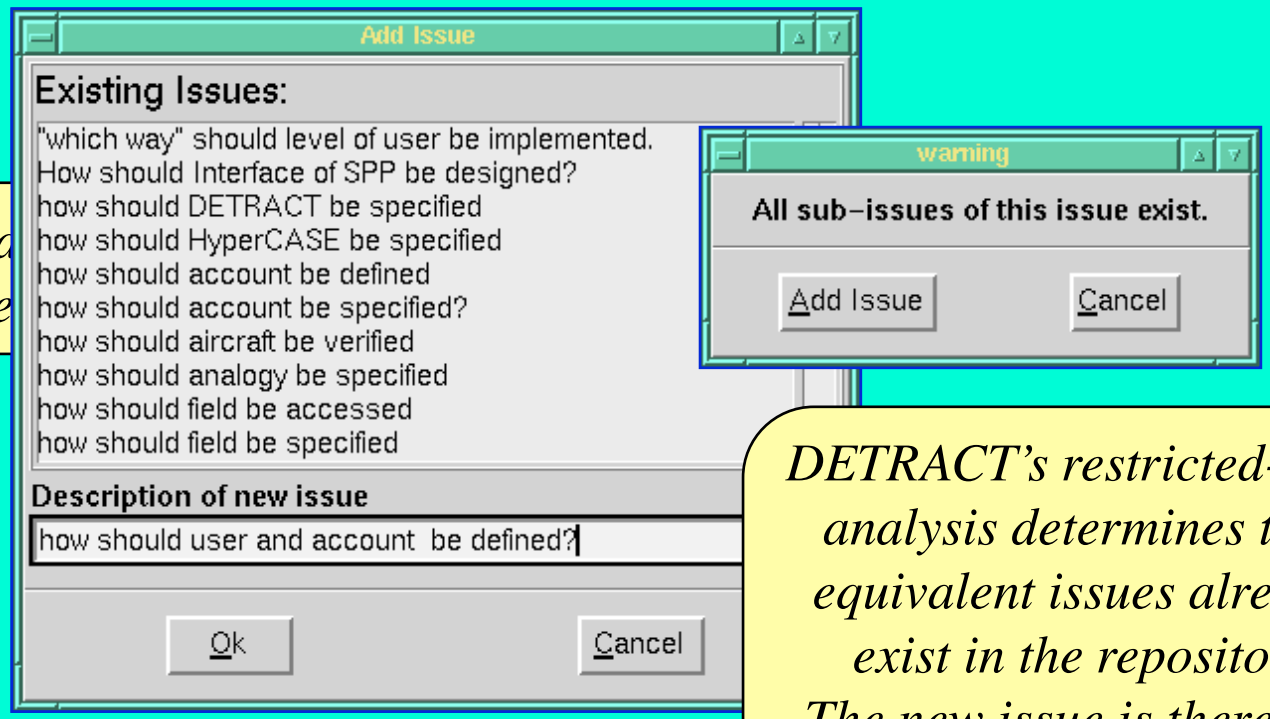
- Example *(an alternative)*

```
Use Screen Section in Sun Cobol
  to implement UI of NBS.
```



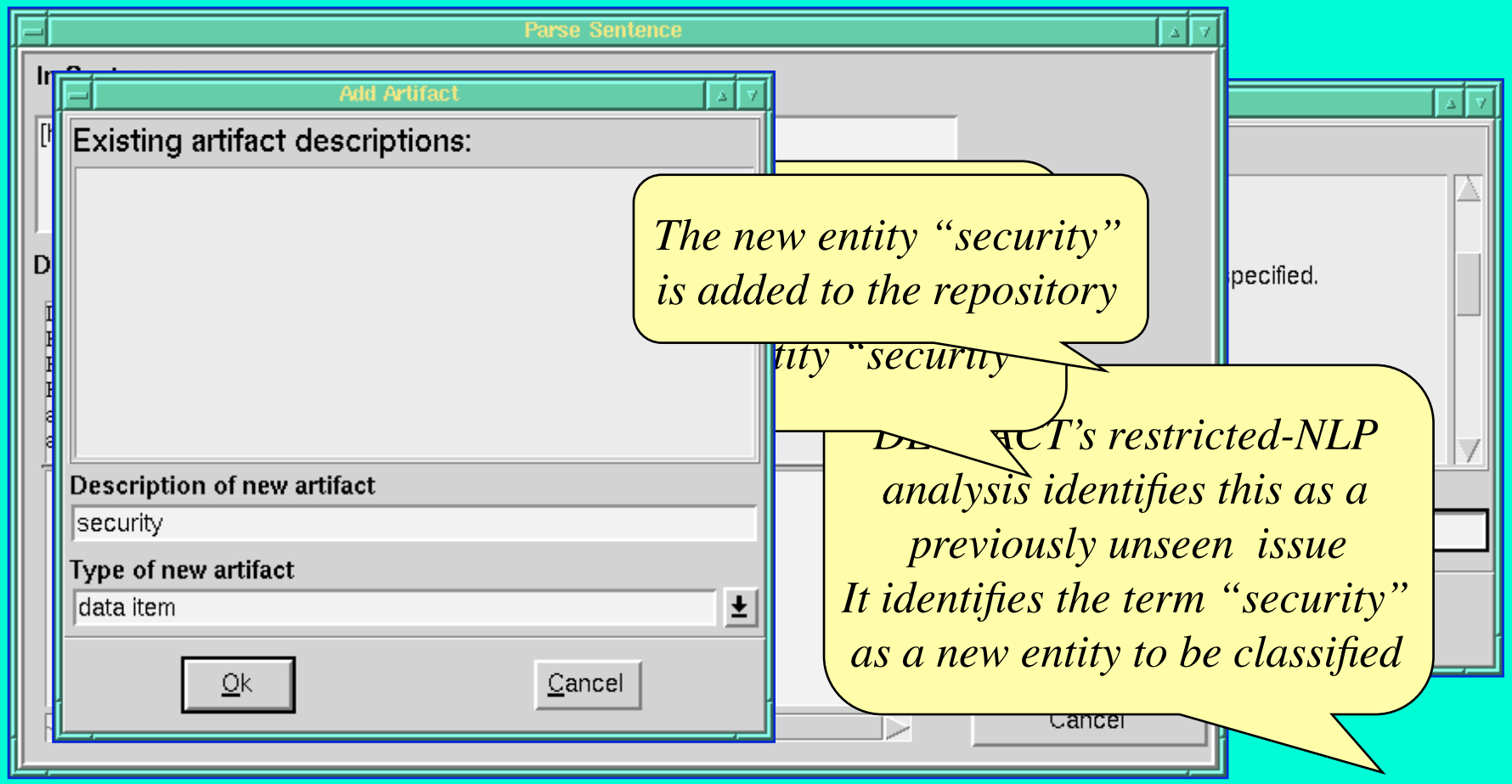
# Recording a New Design Decision Element

*The user adds a new issue*




*DETRACT's restricted-NLP analysis determines that equivalent issues already exist in the repository*
*The new issue is therefore not added*

# Recording a New Design Decision Element



detract

File Search Move Edit Help Options

**Artifact:** query user
**Type:** rule
**Actual:** I:/aaitp/ostar/david:user_on_q

**Change History:**
31 Oct 1996 Karl Reed (kreed)
31 Oct 1996 Abou–Torab Torabi (torab)
31 Oct 1996 Abou–Torab Torabi (torab)

**Issues** Add Del
how should priority of account be mc
how should security be defined?
how should security be implementec
how should status of user be modifie
how should type of user be verified?
how should user be defined?

**Sub Artifacts** Add Del
check user logon
logged–on
set not logged–on
User logon Status

**Goals** Add Del
count user logons
hide user status

**Constraints** Add Del

*new issue*

*Relations to the newly added issue and the entity"security" are added to the "query user" artefact*

he user logon status and then check the actual

*new relation*

**Related** Define Del
[detract ](relating to ) security
[ted ](Depends To ) I:/aaitp/ostar/d
[ted ](Depends to ) I:/aaitp/ostar/da
[detract ](relating to ) user

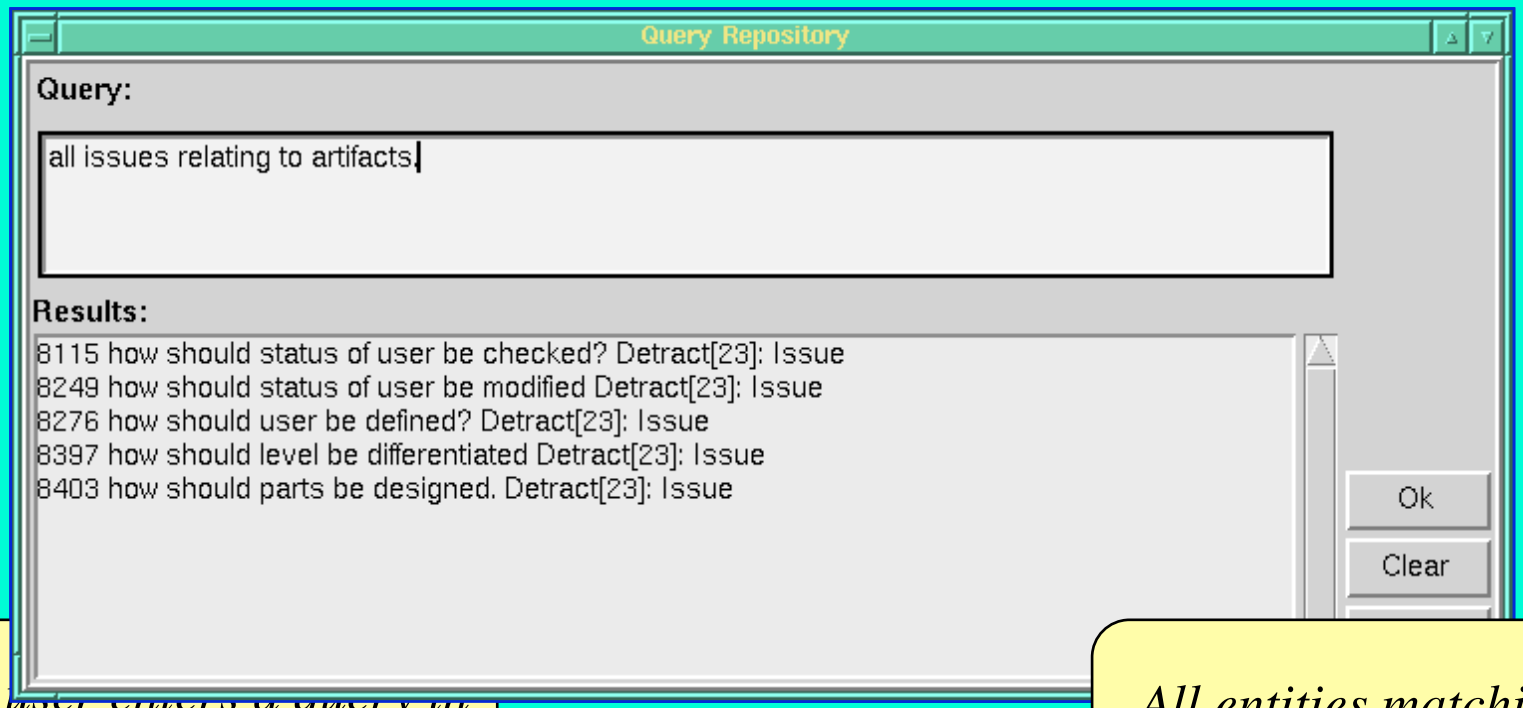
PROJECT (HyperCASE for Object Star) ISSUE (how should security be defined?)

# Querying

- What kind of decisions have been made?
- What kinds of design decisions and design artefact relations exist?
- What other decisions are to be made?
- With what priority should other decisions be made?

# Querying

- Queries
  - retrieve information from the repository
  - traverse network of entities and relations *(closure operations)*
- Entities
  - design decision elements
  - design artefacts
- Relations
  - generic / relationship types / attribute-value
  - implied / explicit




*The user enters a query in terms of entities and relationships from the repository*

*All entities matching the query are retrieved*

# Current Implementation - Summary

- Provides a structured view of the dialog surrounding software projects
- Can be used interactively during software development
- Operates at various levels of granularity
- Allows cross-referencing to any kind of external document
- Supports queries

# C.AAITP Compilable Restricted natural Language (CRNLP) approach, (Controlled Languages)

Objectives….

1. Make “NL” analysable by machine (CRNLP)
2. Make “NL” readily unambiguous in a specific domain
   - E.G. Aircraft maintenance standard English, Caterpillar controlled language (CL)
3. Provide precise data extraction from “NL”
   - E.G. Canonical representations capable of being stored and compared
4. Allow identification of “identical”, “similar” things in “NL”
   - E.G. Requirements fragments, for re-use (SODA), Design Reasoning Recording
5. Capture of domain knowledge and construction of thesaurus

## C. AAITP Compilable Restricted natural Language (CRNLP) approach, (Controlled Languages)(cont'd)

A lot of activity, for quite some time… some of it probably instinctive.

# AAITP Example..(cont'd)

Goal.. To prevent confusion when the following has been written

§1 **Insert** ~~*Add*~~ *the customer_name to the customer list*

…..

§1023 *Insert customer_id in the customer table*

….

§1592 *Copy customer_id to customer_list*

….

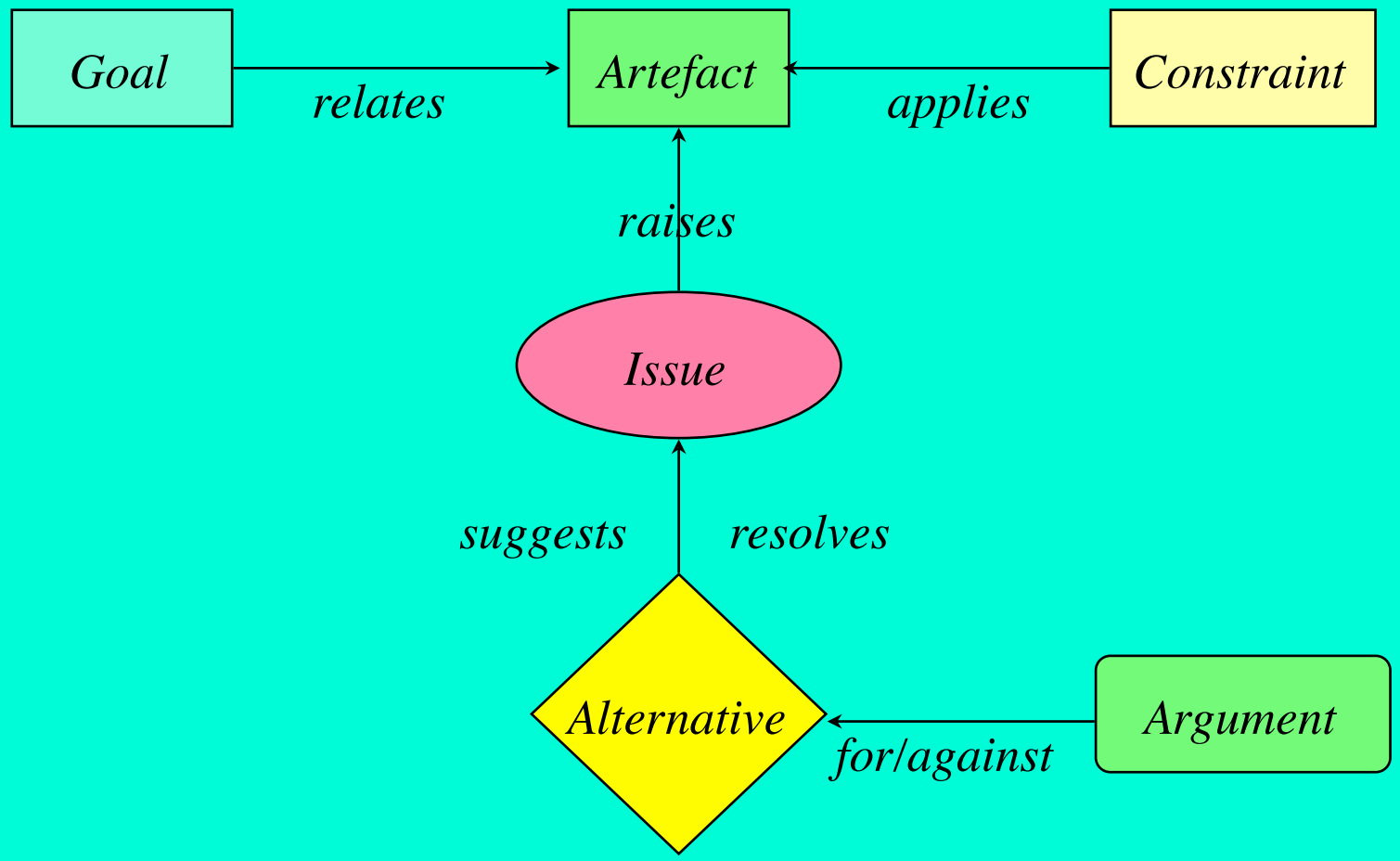
§2437 **Insert** ~~*Put*~~ *the customer in customer file*

…

This might be new, so, add it to thesaurus,else change it!

These three are probably the same, so MAKE them the same!

# Design Decision Model (Again..)



# Example of CRNLP Grammar for DETRACT(TR040, Torab, 1996)

**4.2 Issue**

An Issue or Problem is a statement describing the design problem raised during some stage of the design. An example for an issue could be “How should HyperEdit and Link Server be integrated?”.

<Issue>::=<Qword><modal><NP1> 'be'<pp-verb> ‘?’
<Qword>::= {*how, which way*}
<modal>::= {*should, could, would, can*}
<pp-verb>::= {*implemented, designed, ...*}
{ *This is a list of verbs which can grow, see comments for goal*}

examples:

*How should HyperEdit and Link Server be integrated?*
*Which way should Links in HyperText be added?*
*How should user-interface of DETRACT be implemented?*

# Example of CRNLP Grammar for DETRACT(TR040, Torab, 1996)

**4.3 Alternatives**

Once an issue is raised during the design, designers may propose different alternatives which they believe could resolve the issue. The objective is to choose the best alternative from all those proposed which would resolve the issue, and would satisfy the design goals.

<Alternative>::= <Verb1><NP1><VP1>'.'

<VP1>::='*to*' <Verb><NP1>

<Verb1>::={*use, have, select*, ...}

{ This is a list of verbs which can grow, see comment for goal}

<Verb>::={*specify, implement*, ...}

{ This is a list of verbs which can grow, see comment for goal}

examples:

*Have menu to Specify Link in a tool.*

*select document to specify link .*

*use dialog-box to specify link.*

*Use TCL/Tk to implement user-interface of DETRACT .*

# Example of CRNLP Grammar for DETRACT(TR040, Torab, 1996)

**4.4 Argument**

With respect to each alternative people may raise arguments, statements, facts, and speculations pro or against the alternative. An example of such an argument would be a screen in HyperEdit has an attribute.

<Arguments>::=<NP1><VP>'.'
<VP>::= <Verb><Adj-ph>
<Verb>::={is, has, ...}
{ This is list of verbs which can grow, see comment for goal}

examples:
*Target of Link is difficult.*
*Source of link is not difficult.*
*a screen in HyperEdit has an attribute.*

An argument may support or deny an alternative.

# Example of CRNLP Grammar for DETRACT(TR040, Torab, 1996)

**4.5 Constraint**

Constraint is a statement specifying the restriction should be considered during the design. It again specifies attributes or features of the system which must met during the design.

<Constraint>::=<NP1><VP><PP0>'.'
<PP0>::=<prepos><NS>
<VP>::=<modal>'be'<pp-verb>
<prepos>::={*in, using, by*}
<model>::={*should, must*}
<pp-verb>::={*implemented, specified*, ...}
{ This is a list of verbs which can grow, see comment for goal}

examples:

HyperEdit must be implemented in tcl/tk.

Links in HyperEdit should be specified by dialog-box.

# D. AAITP HyperCASE framework

**AAITP TECHNICAL** *are based on a number of ideas…*

***a.*** *The design-distance between system concept and implementation should be as short as possible…*

*gn must be as informative as pos*

*ming tools linked to implementati*

*ecific visualisation, graphical c*

***s must be linked for ease of management…***

***(i.e. cross referencing between "ObjectStar rules & tables, other code fragments, diagrams, SRS, etc., powerful query support for component and document recovery and navigation)***

***d. Re-use at the highest levels is essential.***

***e. Design processes should be recorded in a re-playable manner***

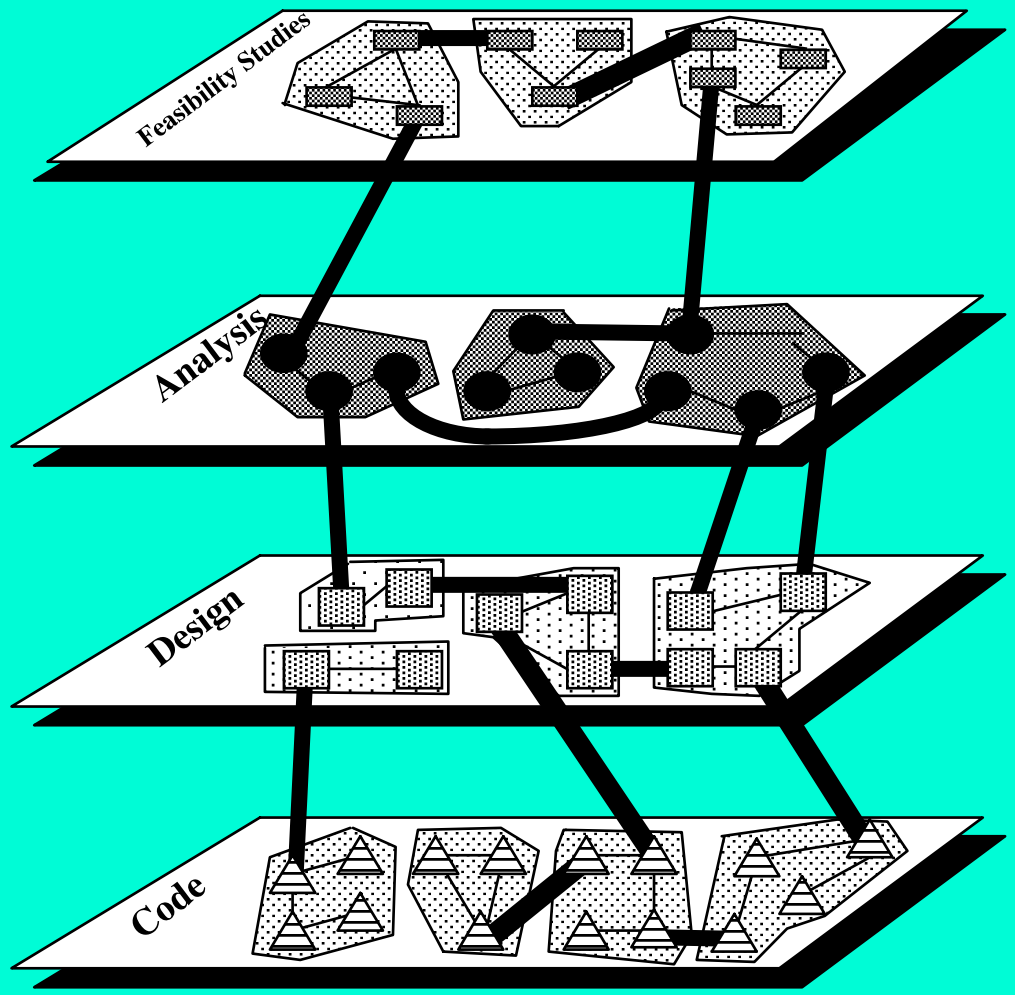
*f. Project progress should be tracked automatically*

*Links between re-engineers "reasons" and a created just drawn of a possible architecture*

*Decision-components can be linked to artefacts that they refer to.*

# A view of project documents

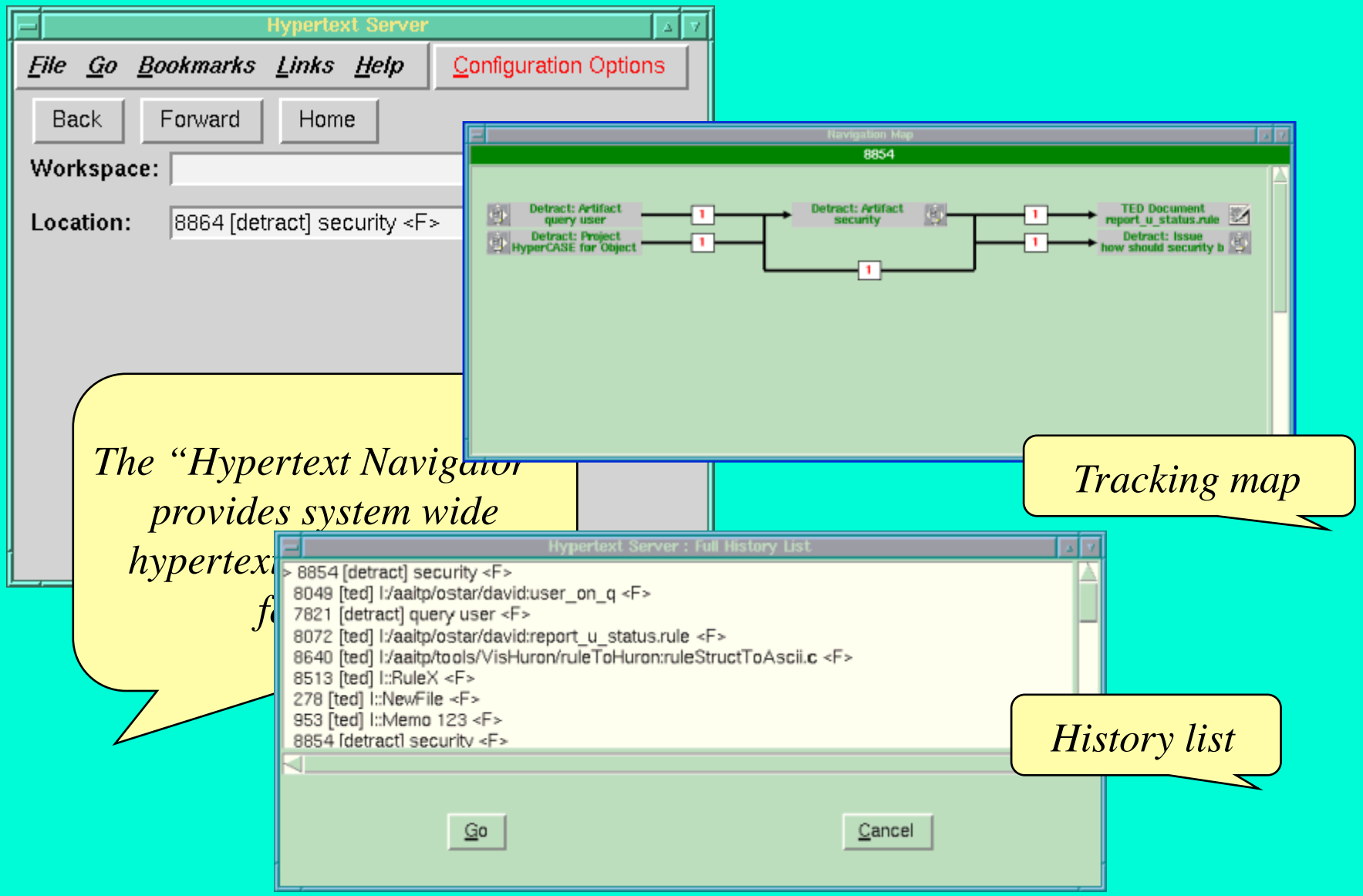
## Searching between planes

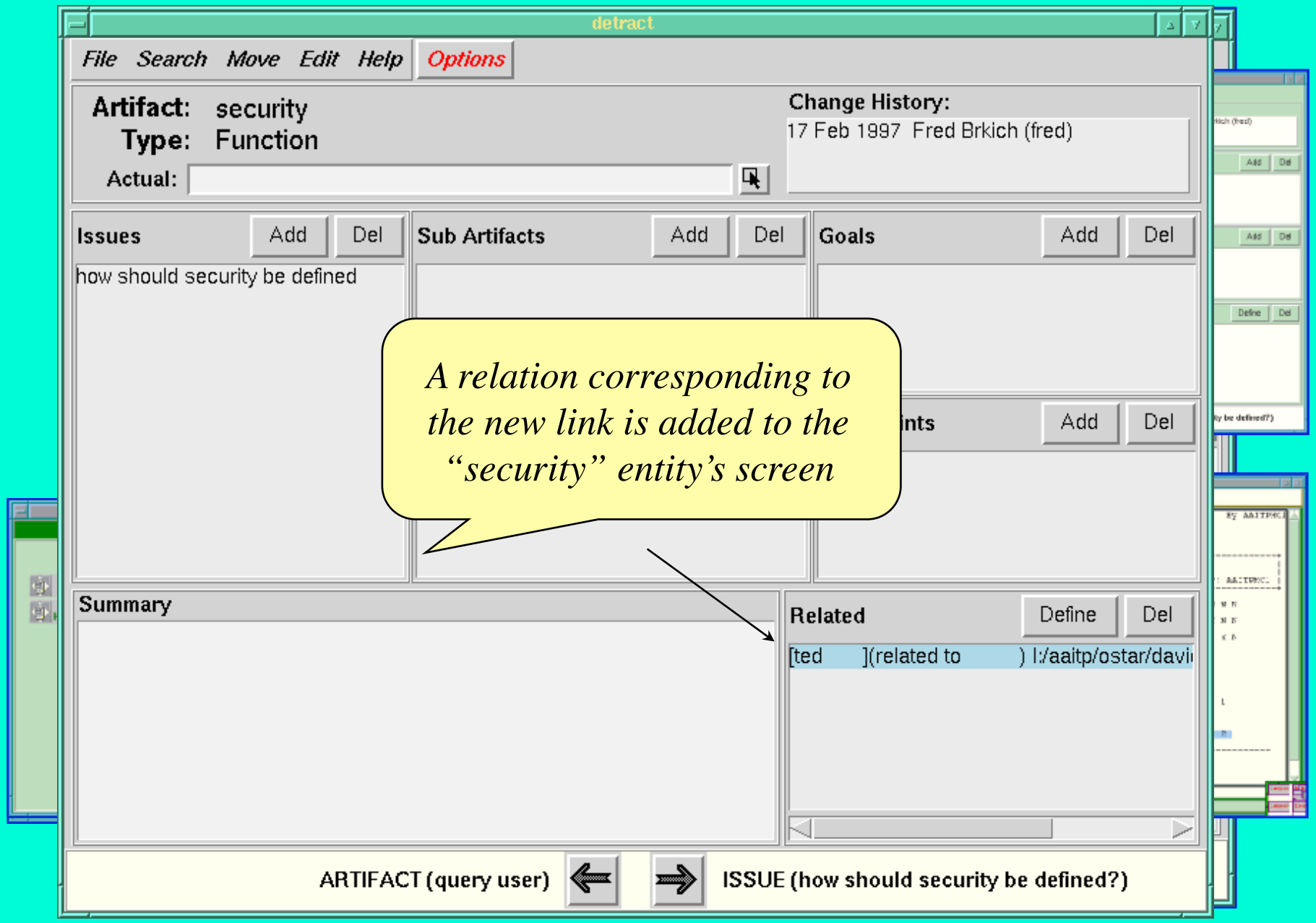


# HyperText Navigation
# In the HyperCASE Environment

# Hypertext Navigation

- Hypertext navigation
  - creation of hypertext links across applications
  - integrate data from a diverse set of applications
  - hypertext traversal internally and externally
  - customisation
- Standard features including
  - history lists
  - bookmarks
  - tracking maps
  - user profiles


Hypertext Server
File Go Bookmarks Links Help
Configuration Options
Back
Forward
Home
Workspace:
Location:
8864 [detract] security <F>
Navigation Map
8854
Detract: Artifact query user
Detract: Project HyperCASE for Object
Detract: Artifact security
TED Document report_u_status.rule
Detract: Issue how should security b
Tracking map
The "Hypertext Navigator
provides system wide
hypertex
f
Hypertext Server : Full History List
> 8854 [detract] security <F>
8049 [ted] I:/aaitp/ostar/david:user_on_q <F>
7821 [detract] query user <F>
8072 [ted] I:/aaitp/ostar/david:report_u_status.rule <F>
8640 [ted] I:/aaitp/tools/VisHuron/ruleToHuron:ruleStructToAscii.c <F>
8513 [ted] I::RuleX <F>
278 [ted] I::NewFile <F>
953 [ted] I::Memo 123 <F>
8854 [detract] security <F>
Go
Cancel
History list




*A relation corresponding to the new link is added to the "security" entity's screen*

# E. Some lines of investigation and possible tool development, Architecture Recovery…

# E. Some lines of investigation and possible tool development, Architecture Recovery…

§ *Conjecture.. During architecture recovery (and re-engineering), DETRACT type recording would be helpful….*

*If so, the approach used in DETRACT can be modified to meet the needs of architecture recovery*

§ Lines of investigation….

LOI1. Review literature and status of CRNLP and Controlled langauges

LOI2. Attend EAMT/CLAW 2003

LOI3.1 "Mine" IESE projects to see how Arch. Recovery is being done.. What do people record, how do they use the tools, and what would they like?

LOI3.2 "Mine" literature for reports on actual mechanisms of re-engieering, not the tools used, what people actually do..

LOI4. Develop some CRNLP prototypes

LOI5. Design/Develop an integrated Hypertext linkage system similar to HyperCASE so that the document collection can be linked to the recording system

# E. Some lines of investigation and possible tool development, Architecture Recovery…

§ Lines of investigation (cont'd)….

LOI6. Develop some collaborative links with the Controlled Language community

# E. Some lines of investigation and possible tool development, Architecture Recovery…

§ *Conjecture.. During architecture recovery (and re-engineering), DETRACT type recording would be helpful….*

*If so, the approach used in DETRACT can be modified to meet the needs of architecture recovery*

§ Lines of investigation….

LOI1. Review literature and status of CRNLP and Controlled langauges

LOI2. Attend EAMT/CLAW 2003

LOI3.1 "Mine" IESE projects to see how Arch. Recovery is being done.. What do people record, how do they use the tools, and what would they like?

LOI3.2 "Mine" literature for reports on actual mechanisms of re-engieering, not the tools used, what people actually do..

LOI4. Develop some CRNLP prototypes

LOI5. Design/Develop an integrated Hypertext linkage system similar to HyperCASE so that the document collection can be linked to the recording system

# Conclusion we've met our Goal!….

A. Describe the Problem,

B. Introduce Design Reasoning Recording, DETRACT

C. Introduce the AAITP Compilable Restricted natural Language (CRNLP) approach, (Controlled Languages)

D. Present the AAITP HyperCASE framework,

E. Suggest some lines of investigation and possible tool development,

*Vielen Dank*

# Bibliography

## *1. Selected Amdahl Australian Intelligent Tool Program TR's*


Cleary, D(1993) *A Frame Based Conceptual Scheme for Tracking Document Development in an Extended Hypertext Environment*, Rev.1.1. TR003, Amdahl Australian Intelligent Tool Program (AAITP), Department of Computer Science and Computer Engineering, La Trobe University, Bundoora 3083, Vic. Australia

Cleary,D, Torabi,(1995)T and Reed,K *Query Sub-System for a Design Decision Tracking Tool*, May 1995, TR038,AAITP.

McLaughlin, A (1991) *Software Requirements Specifications for DETRACT A Software Development Design Tracker*, 1991, TR016, AAITP.

McLaughlin,A and Reed,K(1991) - *Design Rationale: The Missing Ingredient in Software Development*, 1991 TR021, AAITP

Torabi,T(1993) *A Review of SoftEAM and Comparison with DETRACT*, November 1993, TR032, AAITP

Torabi,T(1996) *Design of Natural Language Interface for DETRACT, November 1996,* TR040, AAITP

Cybulski J.L.C. and Reed,K (1992) *A HYPER-TEXT BASED SOFTWARE ENGINEERING ENVIRONMENT* IEEE Software, Vol. 9 No. 2, Mar 1992 pp 62-68

Yong, M, and Reed, K(1992) *Identifying Reusable Components in Software Requirements Specifications to Develop a Natural Language-like SRS Language with a CRNLP*, January 1992, TR005 , AAITP

# Bibliography(cont'd)

## *2. Architecture Recovery Papers*


Bennett, K.H. and Rajlich, V.T. *Software Maintenance and Evolution: a Roadmap* Future of Software Engineering Symposium Limerick Ireland 2000 publ ACM pp73-8

Ding,L. and Medvidovic, N (2001) *Focus: A Light-Weight, Incremental Approach to Software Architecture Recovery and Evolution* Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA'01)

Harris, D.R., Reubenstein, H. B. and Yeh, A.S .*Reverse Engineering to the Architectural Level* ICSE '95, Seattle 1995, ACM pp186-195

Hassan, A.E. and Holt, R.C *A Reference Architecture for Web Servers* Proceedings of the Seventh Working Conference on Reverse Engineering (WCRE'00), Brisbane 20002, IEEE Computer Society

Kazman, R. and Carriere, S.J(1998). *View Extraction and View Fusion in Architectural Understanding* Proceedings of the Fifth International Conference on Software Reuse, Victoria, 1998, IEEE Press

Kruchten, P. (1995). *Architectural Blueprints - The 4+1 View Model of Software Architecture*. IEEE Software (November 1995).

Mendoca, N. C. and Kramer, J(1996) . *Requirements for an Effective Architecture Recovery Framework* Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops October 1996, ACM 1996

# Bibliography(cont'd)

## 2. *Architecture Recovery Papers(cont'd)*


Perry, D.E and Wolf, A.L (1992) *Foundations for the Software Architecture* ACM SIGSOFT Software Engineering Notes Vol. 17 No. 4 Oct 1992 pp. 40-52

Ran,A and Kuusela,J (1996) *Selected issues in architecture of software intensive products* Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops October 1996

Reed, K. (1987). *Commercial Software Engineering, The Way Forward. (keynote address)*. Keynote address Australian Software Engineering Conference, Canberra. ACT. Australia. 1987

Reed,K (2000) *THE FUTURE OF SOFTWARE ENGINEERING AND RE-ENGINEERING AS SOFTWARE ARCHEOLOGY*. Keynote Speech to the 2000 Working Conference on Software Engineering, Brsibane, Nov. 2000

Rugaber, S and Wills L. M. *Creating a Research Infrastructure for Reengineering* 3rd Working Conference on Reverse Engineering (WCRE '96) Monterey November 1996 IEEE Computer Society

Shaw, M., DeLine, R., Klein, D. V., Ross, T. L., Young, D. M. and Zelesnik, G.(1995) *Abstractions for Software Architecture and Tools to Support Them* IEEE Transactions on Software Engineering Vol. 21 No. 4 Apr. 1995 pp 314-335)

Shaw, M. *Large Scale Systems Require Higher-level Abstractions* Proceedings of the Fifth International Workshop on Software Specification and Design, IEEE Computer Society, 1989 pp 143-146

Wileden, J. C. (1986). *This is IT: A Meta-Model of the Software Process. ACM Software Engineering Notes*, 11 (4), 9-11 Proceedings of the International Workshop on the Software Process and Software Environments, Trabuco Canyon March 1986

# Bibliography(cont'd)

## 3. Design Reading Recording Papers


Ambras J., O'Day V. *Microscope: A Knowledge-Based programming Environment*, IEEE Software ,1988, pp 50-58.

Arango G., Bruneau L. *A Tool Shell for Tracking Design Decisions*., IEEE Software, 1991, pp 75-83

Basili V.R. *Viewing Maintenance as Reuse-Oriented Software Development*, IEEE Software, 7(1): pp 19-25, Jan 1990.

Baxter I. D. *Design Maintenance Systems*, CACM, vol 35, no 4, 1992, pp 73-89.

Borgida A., Jarke M. *Knowledge Representation And Reasoning In Software Engineering*, IEEE TSE, vol 18, no 6, l992, pp 449-450.

Card D.N., Church V.E., Agresti W.W. *An Empirical Study of Software Design Practices*, Transaction on Engineering SE 12(2): pp 264-271, Feb. 1986.

Chilkofsky E.J., Cross II J.H. *Reverse Engineering and Design Recovery*: A Taxonomy, IEEE Software, 7(1): pp 13-17, Jan 1990.

Fickas S., Helm R. *Knowledge Representation And Reason In The Design of Composite Systems*, IEEE TSE, vol 18, no 6, l992, pp 470-482.

Freeman P. *Software Design Representation: Analysis and Improvements*, Software Practice and Experience, 8(5): pp 513-528, Sep. 1978.

Freeman P. *Software Design Representation: A Case Study, Software Practice and Experience*, 8(5): pp 501-512, Sep. 1978.

Jayaputea G.T., Cheng K.E. *SoftEAM: A Design History and Justification Maintenance Tool Proceedings* o ASWEC, pp 185-196, Sep 1993.

Kaiser G.E., Feiler P.H., Popovich S.S. *Intelligent Assistance for Software Development and Maintenance, IEEE Software*, 1988, pp 40-49.

# Bibliography(cont'd)

## *3. Design Reading Recording Papers*


Kozaczynski W., Maciaszek *Design Recording As Integral Aspect of Software Engineering*, ASWEC 90, pp 87-92.

Kozaczynski W. *The 'Catch-22' of Re-Engineering*, IEEE ICSE, Nice, France 1990, pp 119.

Lee J. *Extending The Potts And Bruns Model For Recording Design Rationale*, IEEE 1991

Letovsky S., Soloway E.*Delocalised Plans and Program Comprehension*, IEEE Software, 3(3): pp 41-49, May 1986.

Potts C., Bruns G. *Recording The Reason for Design Decision*, ICSE, IEEE 1988, pp 418-427.

Ramesh B., Dhar V.*Supporting Systems Development by Capturing Deliberations During Requirements Engineering*, IEEE TSE, vol 18, no 6, l992, pp 498-510.

Rich C., Feldman Y. *Seven Layers of Knowledge Representation and Reason in Support of Software Development*, IEEE TSE, vol 18, no 6, l992, pp 451-469.

Rich C., Wills LAM. *Recognising a Program's Design: A Graph-Parsing Approach*, IEEE Software, 7 (1): pp 82-89, Jan 1990.

Rugaber S., Ornburn S.B., LeBlanc R.J.*Recognising Design Decisions in Programs*, IEEE Software, 7(1): pp 46-54, Jan 1990.

Schneidewind N. *The State of Software Maintenance*, IEEE TSE ,vol 13 ,no 3, 1987.

Setliff D., Rutenbar R. *Knowledge Representation And Reasoning In a Software Synthesis Architecture*, IEEE TSE, vol 18, no 6, l992, pp 523-533.

Silverman, B *Software Cost and Productivity Improvements: An Analogical View* May 1985 IEEE Computer Vol. 18 No. 5 pp 86-95

# Bibliography(cont'd)

## *4.CRNLP Papers and Sources*

CLAW-EAMT/CLAW 2003 *Controlled Language Translation* Joint Conference of the 7th International Workshop of the European Association for Machine Translation and the 4th Controlled Controlled Language Applications Workshop

Heinrich E., Kemp E. and Patrick J.D. (1999). *A Natural Language Like Description Language*.(Eds.). 10th Australasian Conference on Information Systems (ACIS) Wellington, New Zealand. P. 375 - 386.

Karlgren, Hans (1988)*CONTROLLED LANGUAGES AND LANGUAGE CONTROL* Proc. COLING 88

Tenni,J (1998) *The Webtran Approach* Proc CLAW 1998(?)