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

Karl Reed\(^1\) and Torab Torabi\(^2,3\)

\(^1\)Gastwissenschaftler, 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
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…)

12/2/2006 rev 4
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

12/2/2006 rev 4
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.

Note:
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 loading w.t.

floor \( w_a = 0.15 \times 3.50 \) = 0.53 \( \text{kl} \)

\( w = 0.25 \times 3.50 \) = 0.88

Calc. = 0.10 \times 3.50 = 0.35

front wall:

floor \( w_a = 0.40 \times 4.30 \) = 1.72

\( w = 1.50 \times 2.50 \) = 3.75

\( w = 3.0 \times 1.80 \) = 5.40

For existing beam = 0.20 \( \text{kl} \)

loof \( w_a = 0.15 \times 1.50 \) = 0.23

\( w = 0.25 \times 1.50 \) = 0.38

Calc. = 0.10 \times 1.50 = 0.15

\( \frac{14.08}{14.08} \text{kl} \)

max wt = 30.69 \( \text{kl} \)

Ft. wt. \( \text{on} w_a \) = 7.10 \( \frac{2.5 \times 3.68 \times 4.8 \times 4800}{12.0} \) = 2.89 \( \times 10^{-1} \)
$$EI_{used} = 4.35 \times 10^{12}$$

for existing 2"-270 x 75 Oregon beam

$$EI = 9100 \left( 118 \times 10^3 \times 10^5 \right) = 2.149 \times 10^{12}$$

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

for M.S. Section $$E = 2 \times 10^5$$

fuy = $$180 \times 75$$ parallel flange channel

$$EI = 2 \times 10^5 \left( 14.10 \times 10^3 \right) = 2.82 \times 10^{12}$$

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

max moment $$= 165 \left( 157 \times 10^3 \right) + 9.6 \times 1.43 \left( 85 \times 0.47 \times 10^3 \right) 2 = 4.6 \times 10^3$$

> 30,000 lb

Adopt - existing 2"-270 x 75 Oregon beam

+ additional 190 x 75 Parallel flange channel

1" X 10 bolt at 600 = diagonally staggered try to be better.
max reaction = (69.61 in) internal
(16.19 = PL).

max fact. height = 3000

for 150 x 150 F7 Oneway panel
A = 21610

\[ S_a = \frac{0.35(3000)}{150} \]
\[ = 17.3 \]
\[ P = 0.95 \]
\[ \delta_{op} = 16.15 \]

\[ f_a = \frac{5.10 \times 1.40 \times 0.82}{21610} = 4.50 \text{ MPa} \]

\[ f_c = \frac{69.61 \times 10^{3}}{21610} = 3.22 \text{ MPa} \]

Adopt:
150 x 150 F7 Oneway panel (internal)
support part

Provide min.
600 x 600 x 350 depth

Note:
For end support
110" end beam to beamward
- consider min. design requirements to existing
  - for increased span near

max. span = 24'00' - contin. (2200 + 2000).

max. loading w =

Roof
\[ w = 0.15 \times 5.0 = 0.75 \text{ lb} \]

Ceiling
\[ w = 0.10 \times 5.0 = 0.50 \text{ lb} \]

Floor
\[ w = 1.50 \times 3.20 = 4.80 \text{ lb} \]

beam w = 0.20 lb \[ W_1 = 9.68 \text{ lb} \]

Roof
\[ w = 0.75 \text{ lb} \]

Ceiling
\[ w = 0.50 \text{ lb} \]

Floor
\[ w = 1.80 \times 4.50 = 8.10 \text{ lb} \]

beam w = 0.20 lb \[ W_2 = 12.14 \text{ lb} \]
\[ \text{Load} = 7.14 \times 10^5 \times 2.50 = 17.85 \times 10^5 \text{ lb} \]

Based on observations:
- Existing # = 270 x 75 Oregon beam
- Insufficient

Provide min support:
- 100 x 100 F 7 Oregon post - studwall
- 110 - end bearing to Studwall - not needed
- Critical load: Oregon post
  400 x 400 x 350 --depth core post
RETAINING WALL:

\[ P_a = 2.9700 \]
\[ E = 0.4900 \]
\[ A = 33.9907 \]
\[ F = 0.4500 \]
\[ SAFETY\ FACTOR = 1.8219 \]
\[ e = 0.1128 \]
\[ Y = 0.1122 \]
\[ BEARING = 91.5575 \]

AID TO:

- 270 m cavity backwall
- 12 hours 400 m/2 x 800-kg vert
- 450 mm x 375 mm depth core steel foot (1 layer F8 for 4th minimum)
The result of an engineering design

RETAINING WALL DETAIL.

NOTE:
PROVIDE MOISTURE BARBER AND DRAINAGE BEHIND WALL TO COMPLY TO COUNCIL REQUIREMENTS.

USE 18 1/4:3 MORTAR FOR WALL CONSTRUCTION.
NOTE: ACCORDING TO DIAMOND VALLEY COUNCIL -
SOIL IS ASSUMED - STABLE-INTERMEDIATE

Min pitch - 2°
Trimdeck roof to match existing - to menu specs - en 50x75 hw
battens @ 1220 crs max
on 12mm w/ pine lining
on 150x22 f7 oregon
beams @ 450 crs (or
150x50 f7 oregon - if
preferred)

19mm 1st floor
on 100x50 hw floor
Joists @ 480 crs
on anteced on 100x75
hw bearers @ 1220 crs
with max span 1500mm

150x50 f7 oregon
post in 600x600x350
deep conc pad -
refer ENG DRWG & COMPS

100x100 rc stumps
on 300x300x150
cong sole plates
@ 700mm depth
NOTE: stumps over
1.350 to be braced

100x:100 f7 oregon
post in 400x400x350
deep conc pad

270 thick brick
retaining wall-
brick pliers @
1200 crs-
520x375 deep
conc footing
with 16m top &
bottom - 4 main
wires - 65mm
cover - refer
ENG DRWG & COMPS

Draftcorp
P O Box 404, Greensborough, 3088
tel: (03) 485 2208, fax: (03) 484 1670

PROPOSED ALTERATIONS & ADDITIONS
TO EXISTING B/V HOME

for: MR K & MRS R REED
at: 150 RYAN'S RD,
ELTHAM NORTH
LOT 3

12/2/2006 rev 4
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!

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.
A. Describe the Problem,

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 language.. “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 languages
B. DETRACT
A Design Decision Tracking and Reasoning Tool

Torab Torabi
Jamie Lenehan
Fred Brkich
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
The DETRACT Tool

DETRACT's initial screen lists all projects in the repository. Select project "HyperCASE for ObjectStar." All the project's artefacts are displayed. 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
[ted] (Depends To) l/aaltp/ostar/davici/user_on_q
[ted] (Depends to) l/aaltp/ostar/davici/user_on_q
[detact] (relating to) 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 Environment and Relationship between Design Artifacts

12/2/2006 rev 4
Design Decision Model

Goal relates Artefact applies Constraint

raises

suggestions

resolves

for/against

Alternative

Argument

12/2/2006 rev 4
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.
The user enters a second new issue. DETRACT’s restricted-NLP analysis identifies this as a previously unseen issue. It identifies the term “security” as a new entity to be classified. The user enters type information for the new entity “security.” The new entity “security” is added to the repository.
Relations to the newly added issue and the entity "security" are added to the "query user" artefact.
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
    \textit{(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 Add the customer_name to the customer list

....

§1023 Insert customer_id in the customer table

....

§1592 Copy customer_id to customer_list

....

§2437 Put the customer in customer file

...

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

This might be new, so, add it to thesaurus, else change it!
Design Decision Model (Again..)

- **Goal** relates to **Artefact**
- **Artefact** applies to **Constraint**
- **Issue** raises
- **Issue** suggests **Alternative**
- **Alternative** resolves** Argument

**Note:**
- **For/against**
- **Suggests**
- **Resolves**
- **Applies**
- **Relates**
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?”.

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

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?*
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.
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.

\[
<\text{Arguments}> ::= <\text{NP1}> <\text{VP}> \cdot
\]

\[
<\text{VP}> ::= <\text{Verb}> <\text{Adj-ph}>
\]

\[
<\text{Verb}> ::= \{\text{is, has, ...}\}
\]

{ 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...

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

Links between re-engineers “reasons” and a created just drawn of a possible architecture 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
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
The “Hypertext Navigator” provides system wide hypertext functions and features.

- Tracking map
- History list
A "point and click" approach is used to define hypertext links between artefacts of various types. A hypertext link is to be defined with the DETRACT entity "security" as its source and the ObjectStar rule "REPORT_U_STATUS" as its destination.

The tracking map shows the current hypertext links in and out of entity "security". "Clicking" on entity "security" pops-up the link definer with the source already identified. "Clicking" on rule "REPORT_U_STATUS" identifies the destination. The user completes the relation information, and the hypertext system is automatically updated including showing the new link on the tracking map. 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 languages
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-engineering, 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

12/2/2006 rev 4
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 languages

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-engineering, 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.


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
2. Architecture Recovery Papers


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


Bibliography (cont’d)

2. Architecture Recovery Papers (cont’d)


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


12/2/2006 rev 4
Bibliography (cont’d)

3. Design Reading Recording Papers


12/2/2006 rev 4
Bibliography (cont’d)

3. Design Reading Recording Papers

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


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


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

