Case Representation

from

Experience Management (chapter 3)

– Ralph Bergmann

Three approaches

- Textual CBR
  - Cases are text documents
  - useful where information exists in text form
- Conversational CBR
  - Case base is a tree of questions
  - a support tool for call center operators
- Structural CBR
  - Cases are composed from a well defined vocabulary of attributes and values
  - complex problems represented systematically

Textual CBR

- Experience is recorded as free text
  - for example in product descriptions
  - service reports
  - trouble shooting records
  - frequently asked questions
- know-how exists in documents
- user can directly make use of the experience contained in the document
- keyword search is the most straightforward
  - new techniques emerging now
- Works well when there are not too many cases, and when each has a short discriminating description

Example of a Textual Case

Frequently Asked Question 241

Title: Order numbers of CPUs with which communications is possible

Description

Question: Which order numbers must the S7-CPUs have to be able to run basic communications with SFCs ?

Answer: In order to participate in communications via SFCs without a configured connection table, the module concerned must have the correct order number. The following table illustrates which order number your CPU must have to be able to participate in these S7 homogeneous communications.

Lesson / solution

Conversational CBR

- Capture the knowledge contained in customer/agent conversations
- list of questions varies from one case to another
- case author decides order of questions
- questions organized in a tree like structure – manually
- useful in domains in which a high volume of simple problems must be solved
- maintenance costs are high

Example of a Conversational Case

Case 241
Title: Printer does not work in the new release

Q1: What kind of problem do you have ? Printer Problem
Q2: Does the printer perform a self-test ? Yes
Q3: Does the printer work with other software ? Yes
Q4: Did you just install the software ? Yes
Q5: Did you create a printer definition file ? Yes
Q6: What release did you install ? 4.2

diagnosis

Problem: Installation procedure overrides printer definition
Action: Reinstall the printer from disk 2.3

therapy

The Structural Approach

- cases expressed with common vocabulary
  - flat attribute-value tables
  - object oriented representations
  - graph structures
  - formulas in first order logic
- additional knowledge must be applied to produce good results
  - knowledge of similarity and retrieval methods
- maintenance effort is low.

Example of a Structured Case

Reference : AD8009
Price : 2.25
Input Offset voltage : 2mV
Input bias current : 50 uA
Output voltage : 1.2 V
Output current drive : 175 mA
Single supply : No
PSPS : 70 dB
Number of devices per package : single
Available Package(s) : SOIC

Recommender systems

Differences between the three deal with:

- the material required to set up the initial case base
- effort required to maintain case base
  - tune performance
  - add new cases
- the effort required to control the accuracy of the experience management system

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Case Base	- A case is represented in free-text format. - may be structured according to the headers, but the content of the headers is in free-text form. - The case base is the collection of free texts that may be, for example, in electronic documents that are accessed on the Internet.	-Cases are represented by a list of question and answers. -The list of questions and answers may vary from one case to another. -There is no common data structure.	-The cases are represented according to the vocabulary, which provides a common structure. - assigning values to certain attributes that have been predefined. -The initial case base may easily be extracted from an existing data base

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Case Base	-A case is represented in free-text format. - may be structured according to the headers, but the content of the headers is in free-text form. - The case base is the collection of free texts that may be, for example, in electronic documents that are accessed on the Internet.	-Cases are represented by a list of question and answers. -The list of questions and answers may vary from one case to another. -There is no common data structure.	-The cases are represented according to the vocabulary, which provides a common structure. - assigning values to certain attributes that have been predefined. -The initial case base may easily be extracted from an existing data base

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Query Case	A query is represented by a question in free-text form whose content is *similar* to the description part of the stored cases.	A list of questions and answers following a dialog with the user	The query is also expressed according to the vocabulary. - It is typically a partially filled case description
Results	A list of documents that might be useful.	A list of possible actions attached to the cases that have been retrieved	A list of structured cases, e.g. database records

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Background Knowledge	Dictionaries of similar terms like “install:setup” or “printer:plotter” and information about the relationships between different words	Ordering of questions, hierarchy of questions in a tree-like structure, possible answers for the questions that have been defined.	Vocabulary that defines the case structure. Rules to deduce values. Information on how to compute the similarity among attribute values.

One must tackle small differences in text and still retrieve relevant documents

Similar problems have similar solutions

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Initial Effort	Defining terms, synonyms and stop-words. Analyzing and setting up relationships between different words	Storing dialogs within a case base, structuring the questions to organize the case base manually, ordering the questions manually, This can be done by entering existing decision trees.	Defining vocabulary. Importing an existing database or collecting cases according to the structure of the vocabulary.

Stop words, or **stopwords**, is name given to words which are filtered out prior to, or after, processing of natural language data (text).

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Case Creation	Writing new documents.	Adding new dialogs. Depending on the structure of the dialog. New questions must be entered.	Adding a new record to the database. A questionnaire (forms) can be used in order to use the same vocabulary.

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Maintenance	Maintaining the dictionary of controlled terms and sentences. Analyzing user queries and results	Maintaining the list of questions and answers. Eliminating doubles, combining answers and questions. Reordering the questions by hand for the consultation.	Maintaining the case base, the vocabulary and the reuse-related knowledge.

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Multilingual Case Bases	Almost impossible. Must have different documents.	Difficult. Must have different conversations for each language.	Easy to implement with a single case base. Translated based on the vocabulary.

Recent efforts using machine translation for cross lingual search on the web.

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Advantages	Existing documents can be used as cases. No initial investment is required for modeling the cases.	The approach is intuitive and easy to understand. Appropriate for simple applications.	An existing data base can be used to produce the initial case base. High accuracy can be achieved. Appropriate for complex applications.

Comparison of Different CBR Approaches

Criteria	Textual CBR	Conversational CBR	Structural CBR
Drawbacks	The user is not guided. The quality of retrieval depends on the syntax and not on the real content of the cases. Difficult to achieve high accuracy. High cost for quality control of the CBR system.	No explicit knowledge represented in the system for computing similarity. Interdependencies among different dialogs cannot be predicted. High maintenance costs.	It can be difficult to create a predefined case structure. This can usually be achieved within most technical domains but becomes harder in softer domains. High investment costs.

Efforts Required for Different Approaches

Effort per Task	Textual	Conversational	Structural
Reuse existing material	Very Low	High	Medium

Cases cannot be loaded directly from a database. For each case questions have to be positioned manually by the developer

Efforts Required for Different Approaches

Effort per Task	Textual	Conversational	Structural
Initial Modeling	High	Low	High

Defining terms, synonyms, stop words

Defining database structures, attribute and their values

Efforts Required for Different Approaches

Effort per Task	Textual	Conversational	Structural
Case Creation	Low	High	Medium

Cases cannot be loaded directly from a database. For each case questions have to be positioned manually by the developer

Efforts Required for Different Approaches

Effort per Task	Textual	Conversational	Structural
Tuning CBR	Very High	High/Impossible	Medium-Low

Difficult to tune the system to retrieve the right document. Difficult to maintain this afterwards. New documents may alter the retrieval quality.

There is no notion of similarity. System functions as black box. Developer may have to dummy questions to get around problems

Efforts Required for Different Approaches

Effort per Task	Textual	Conversational	Structural
Quality Control	Very High	High	Low
Maintenance	Low for the case base, High to tune the retrieval	High	Very Low

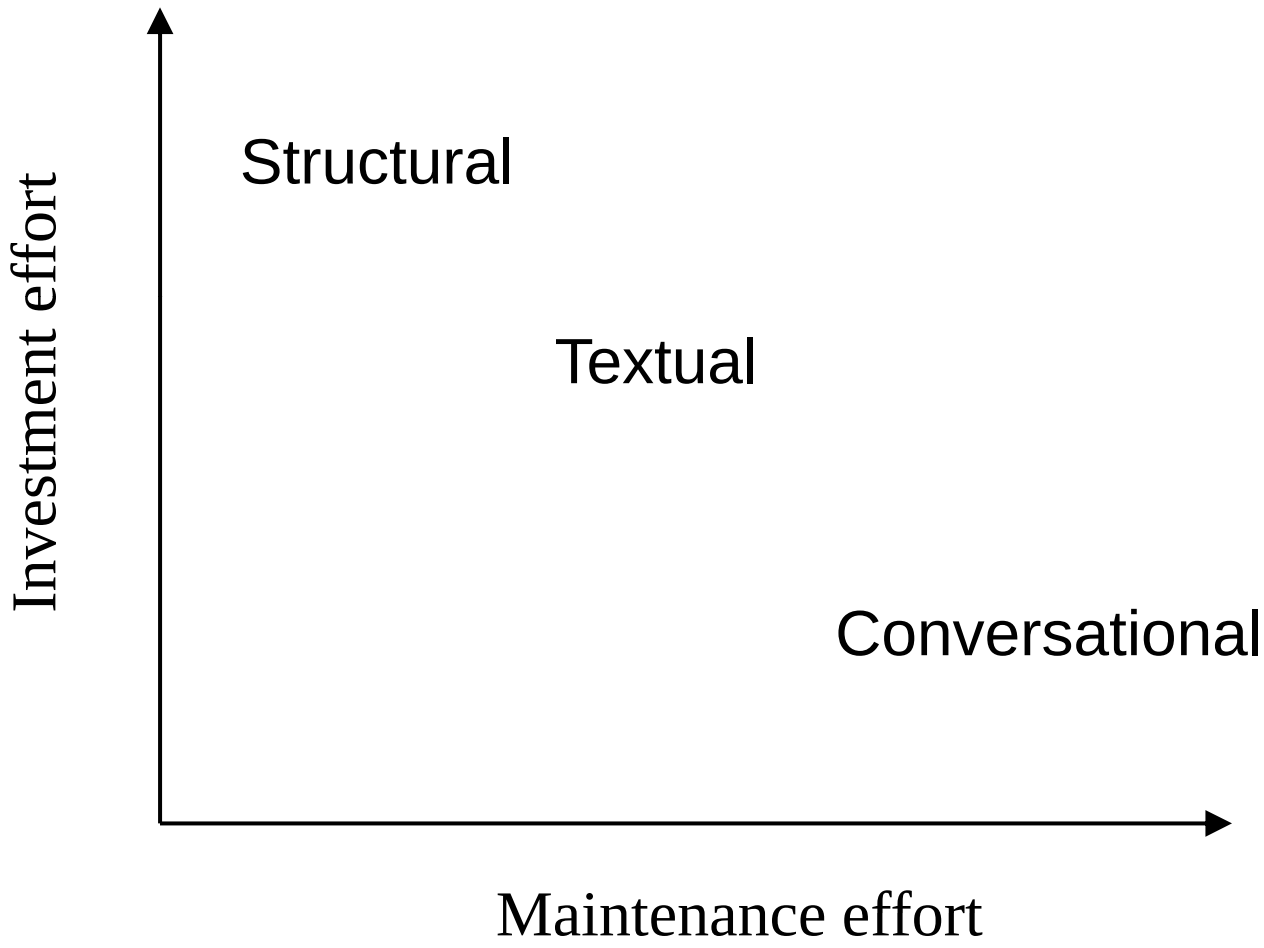
Difficult to tune the system to retrieve the right document. Difficult to maintain this afterwards. New documents may alter the retrieval quality.

Retrieval process may be altered when new questions are defined and entered. Ordering of questions has to be done by author/ developer.

Efforts Required for Different Approaches

Effort per Task	Textual	Conversational	Structural
Reuse existing material	Very Low	High	Medium
Initial Modeling	High	Low	High
Case Creation	Low	High	Medium
Tuning CBR	Very High	High/Impossible	Medium-Low
Quality Control	Very High	High	Low
Maintenance	Low for the case base, High to tune the retrieval	High	Very Low

Overall Comparision



Structural cases

- Cases described in terms of fixed vocabulary
  - set of attributes
  - each attribute has a set of values

- Cases may be
  - flat – collection of attribute value pairs
  - may have some topological structure

Type and Type Space

A *type* T is a pair (T_{name}, T_{range})

T_{name} – A label out of some *type namespace*

T_{range} – A set of values that belong to the type

$a \in T$ denotes a value of type T which is a short notation for $a \in T_{range}$

Type Space is a set of types $\{T_1,...,T_n\}$ with disjoint T_{iname} labels

Examples of Types

- numerical types such as integer or real
- symbol types define by enumeration
- textual types such as strings
- special types such as urls for multimedia objects
- set types defined on some types

Attribute and Attribute Space

An *attribute* A is a pair (A_{name}, A_{type})

- A_{name} – is a label out of some *attribute name space*
- A_{type} – is the label of a type from some type space T

An *attribute Space* is a finite list of attributes (A_1,..., A_n) with disjoint attribute names.

Vocabulary Container

- Vocabulary container VOC in attribute value representations is the pair (A,T) of attribute space and type space

- cases are defined by choosing attributes and values for the attributes from the two respective spaces

- cases may have a fixed set of attributes or each case may have different attributes

Case with fixed set of attributes

- A case with a fixed set of attributes for a vocabulary VOC = (A,T) is a n-dimensional vector (a_1,…, a_n).
- The vector component a_i specifies the value for the attribute A_i in this specific case.
- The value a_i must be an element of the range of the type specified for the attribute A_i

Case with variable set of attributes

A case with a variable set of attributes for a vocabulary VOC = (A,T) is a set of the following form $\{A_{iname} = a_1, \ldots, A_{kname} = a_k\}$

A_{iname} $(i \in 1..k)$ – is the name of an attribute in A

a_i – is an element from the range of the type specified for this attribute.

$A_{iname} \neq A_{iname}$ iff $i \neq j \in 1..k$

Object oriented cases

- Object oriented paradigm
  - is-a and part-of relations
  - inheritance principle
- Object classes
  - arranged in a class hierarchy
  - subclasses inherit attributes as well as their definition
- Attributes may be
  - simple: holding scalar values
  - relational: holding complete objects – represent a part-of relation between two objects

Class Hierarchy

- The *class hierarchy* CL is a finite set of classes $\{C_1, \ldots, C_n\}$
- A *class* C is a tuple
  - $(C_{name}, C_{superclass}, (C.A_1, \ldots, C.A_k))$
  - C_{name} is a label from a class namespace
  - The superclass of root is T
  - $C.A_i$ is a name of an $attribute_i$ of class C from the attribute space A

Extended definition of attributes

- An *attribute* A is a triple
  - (A_{name}, A_{range}, $A_{set\text{-}type}$)
  - A_{name} is name from attribute namespace
  - A_{range} is label from a type space T or a label from the class space CL
  - $A_{set\text{-}type} \in \{true, false\}$ a binary flag that indicates whether the attribute holds a single value or a set of values
- The attribute space A is a finite list of attributes (A_1, …, A_n) with disjoint names

simple attribute

relational attribute

Relations between objects

- The *part-of* relation is implemented by relational attributes
- The *is-a* relation represented by the class hierarchy
- L_C is the set of leaf nodes that lie below the class C
- $C_i < C_j$ denotes that C_i is a successor (subclass) of C_j
- $\langle C_k, C_l \rangle$ denotes the lowest common ancestor of C_k and C_l
  - $\{\langle C_k, C_l \rangle \geq C_k$ and $\langle C_k, C_l \rangle \geq C_l\}$ and there is no class C' such that $\{C' \geq C_k$ and $C' \geq C_l$ and $\langle C_k, C_l \rangle \geq C' \}$

The vocabulary container for OO cases

- The vocabulary container VOC in object-oriented representation is the tuple
  - (CL, A, T, C_{case})
  - of class, attribute and type spaces, and
  - a specifically marked class $C_{case} \in CL$
- C_{case} determines the class of objects that constitute cases.
  - a kind of main object class for the cases

Object Oriented Case

An *object oriented case* c for a vocabulary
VOC = (CL, A,T,C_{case}) is a set of objects c = {o_1,..,o_m}.

An *object* is a tuple
o = (o_{class}, o_{id}, A_{1name} =a_1,…, A_{kname}=a_k})
o_{class} – is a class name from CL
o_{id} – is an object identifier
A_{iname} is an attribute from a class C $\in$ CL with C $\geq$ o_{class}
where A_i is one of the following kind

- single valued
- multi valued
- single-valued relational attribute
- multi-valued relational attribute

possibly inherited from some superclas s

continued

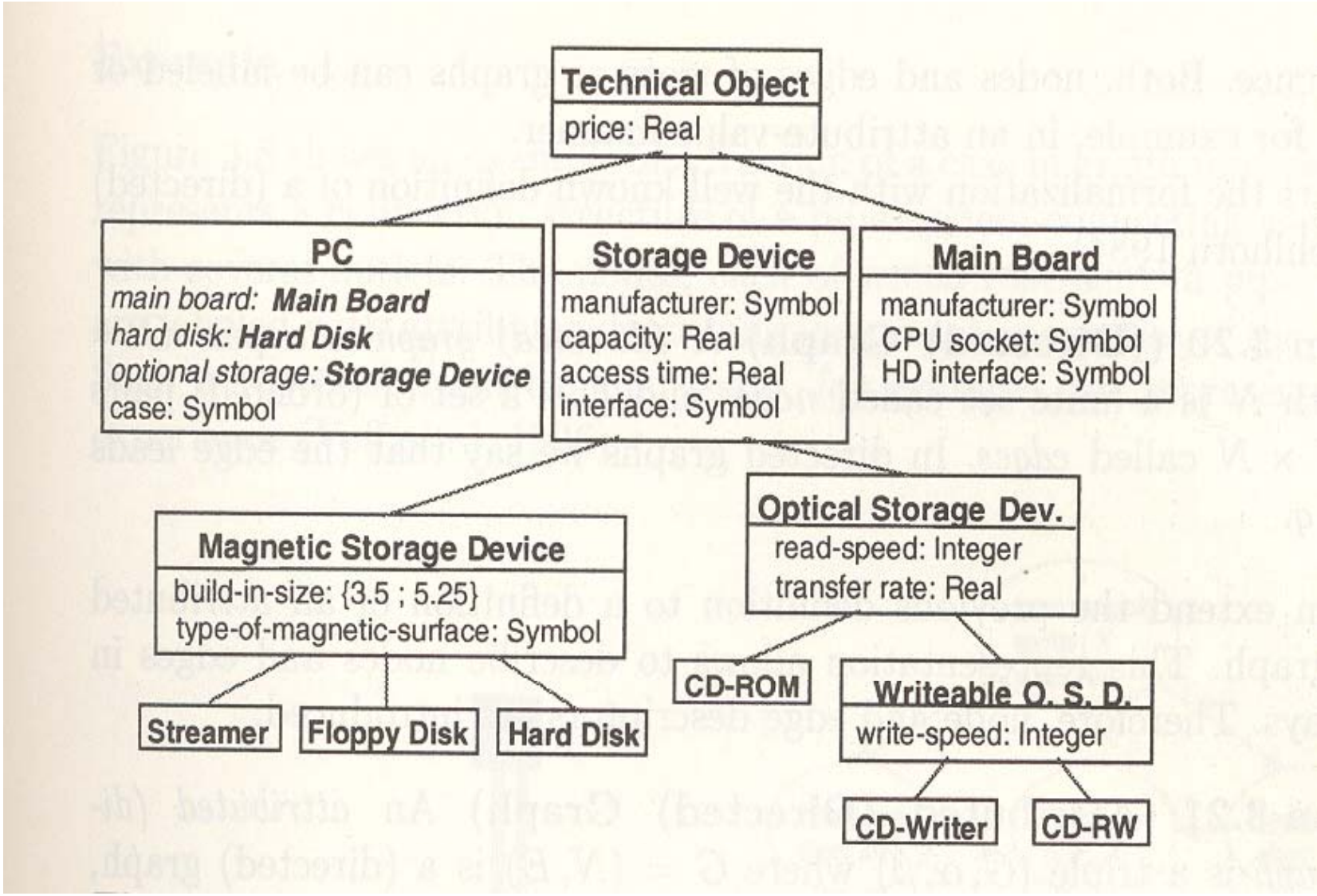
An OO case (continued)

- The a set of objects $c = \{o_1,..,o_m\}$ must contain *exactly one* object o^{case} with

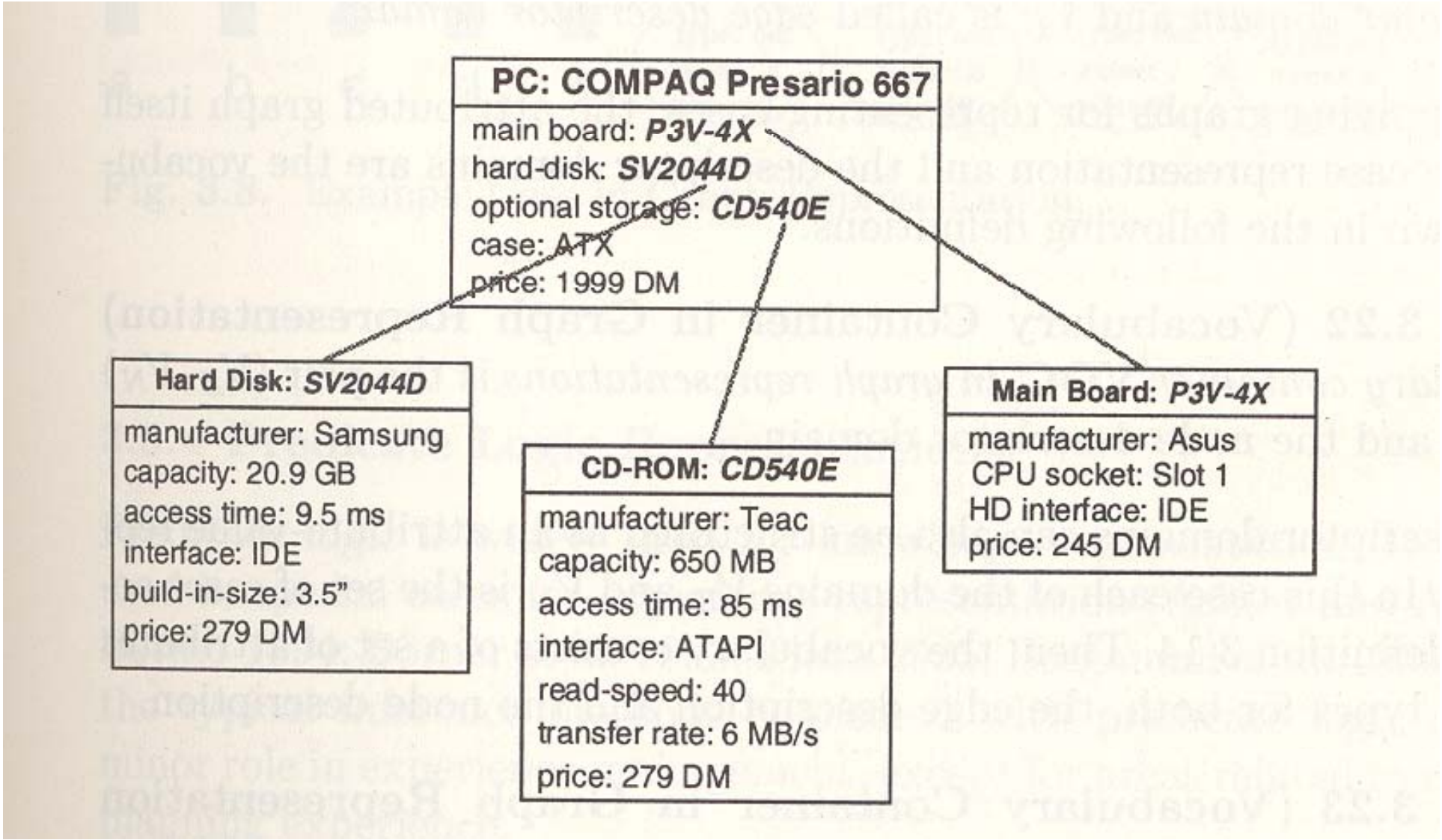
  $o^{case}{}_{class} \leq C_{case}$

- Every other object in $c = \{o_1,..,o_m\}$ must be the value of some other relational object in $c = \{o_1,..,o_m\}$

Class Hierarchy



Example Case



Case in Graph Representation

An *attributed directed graph* is a triple (G,α,β)

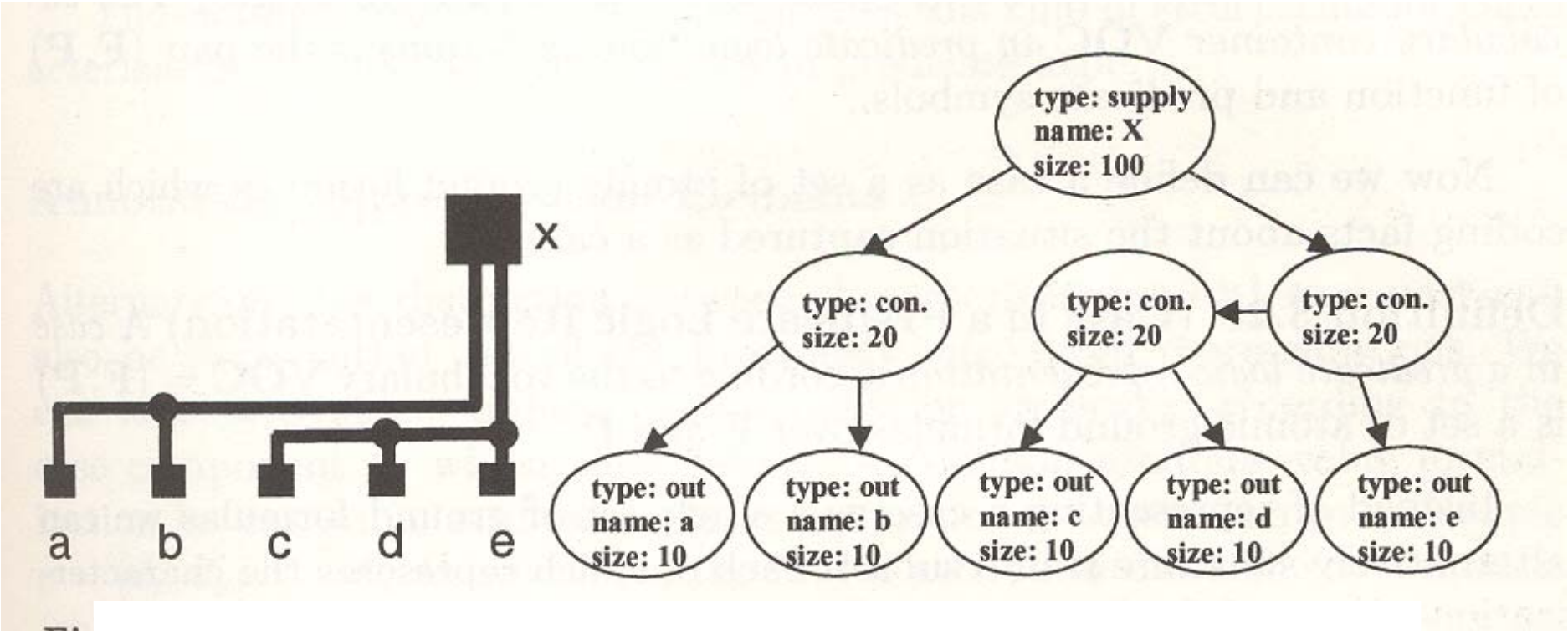
where $G=(N,E)$ is a directed graph, α is a mapping $\alpha:N\rightarrow V_N$, and $\beta:V \rightarrow V_E$.

V_N – node descriptor domain

V_E – edge descriptor domain

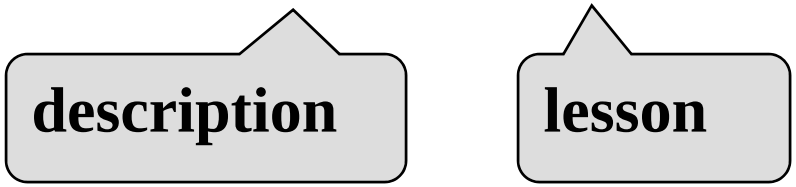
A *case* in a graph representation according to the vocaulary $VOC=(V_E,V_N)$ is an attributed graph or an attributed directed graph (G,α,β)

Graphical Representation



Case in Predicate Logic representation

- Case = (formula1, formula2)



example

- case = ({errorcode(i59), i/o-state(out7,on), relays(rel7, switched), voltage(vdd,gnd, 23.8)}, {fault(magnetic-switch(rack3, msw43))})

Advantages

Attribute - Value	Object Oriented	Graph – Based	Predicate Logic
-Easy to understand and implement -simple and efficient retrieval -link to databases easy	-flexible representation -arbitrary structures can be represented -more compact storage than for attribute-value representations -structural information available for selecting reusable experience	-enables the representation of simple structure -graph algorithms can be applied for the assessment of the relevance of a case.	-very flexible -arbitrary structures can be represented -logical inferences can be used for similarity computation

Disadvantages

Attribute - Value	Object Oriented	Graph – Based	Predicate Logic
-very limited - structural information cannot be represented	-retrieval computationally more complex	-restricted compared to object oriented representations - applicable graph algorithms typically of high complexity	-numeric values can hardly be handled -inferences are computationally of high complexity

Suited for

Attribute - Value	Object Oriented	Graph – Based	Predicate Logic
-simple analytic problem solving tasks - large case bases with a small number of attributes	-complex analytic problem solving tasks -synthetic problem solving tasks, e.g. design and configuration	-analytic and synthetic tasks if networks need to be represented	-synthetic tasks -tasks in which certain rule-like knowledge is dominant for assessing relevance of experience

Inappropriate

Attribute - Value	Object Oriented	Graph – Based	Predicate Logic
-synthetic and complex problem soling	-if no structural information is required	-highly structured domains with different relations	-in domains with many numeric properties -if numeric models for relevance assessment are required