Enough Description of Data and its Utilization

Daisuke Yokouchi (Keio Univ.)

Ritei Shibata (Keio Univ.)

S Project: 1984-1994

Organization of our talk

- Needs of Data and Description
  - NetCDF and DDI
- DandD rule, DandD Instance
- InterDatabase
- DandD Client Server System
  - DandDServer
  - DandDBrowser
  - DandDR
- Data View and Visualization

Needs of Description

- Collaboration between data collector and analyzer.
  - Many people are concerned with a data.
  - on the way from data collection to analysis
- Data Storage
  - Nobody can understand the meaning of the data without enough description, for example, after a year .

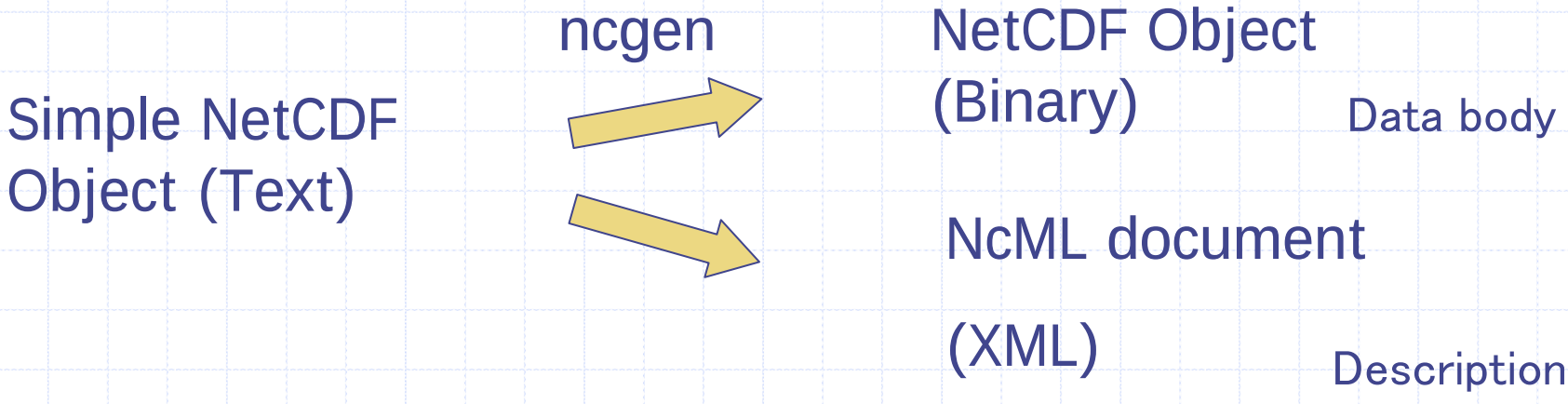
Specific or Unspecific

- Specific Field
  - NetCDF (Unidata Program)
    - For scientific array data
  - DDI (Data Documentation Initiative)
    - For social and behavioral science data
- Unspecific Field
  - DandD (DandD Project)
    - For any data
    - Open ended

NetCDF [Russ Rew et al. (1997) NetCDF User's Guide]

- Network Common Data Form
- CDL (Common Data Language)
  - Notation to describe a data

    → Simple NetCDF Object
- Recent Advances



Problems of NetCDF

- Insufficient data attributes
  - String ? Categorical ? Or Date?
    - Only string
  - Precision ?
    - Only integer, float or double
  - Missing or Invalid ?
    - Only a missing type
- No generarity

DDI [the Data Documentation Initiative(2003), DDI Front Page]

- For social and behavioral science
  - Census data
  - Questionnaire survey data
- Implementation
  - XML

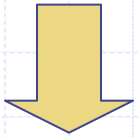
Problems of DDI

- Sampling Design
  - free description
    - No intelligent use of the design
- Data Organization
  - Variable Definition
    - No distinction between Axis and Value

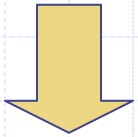
DDI is limited to describing the result of survey.

DandD (Data and Description)

- Philosophy of Data Science
  - Data without description is meaningless.



DandD rule

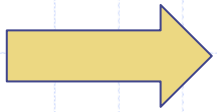


DandD Client
Server System

DandD rule (DDI)

- Element
  - Vector (a sequence of numbers) + Attributes
- Organization of elements
  - Array
  - Relational + Systems
- Background information (Multi-language)
  - Introduction
  - References
  - Relatives (Relation to other DandD Instances)
  - DataSampling

UTF16



DandD instance (an XML document)

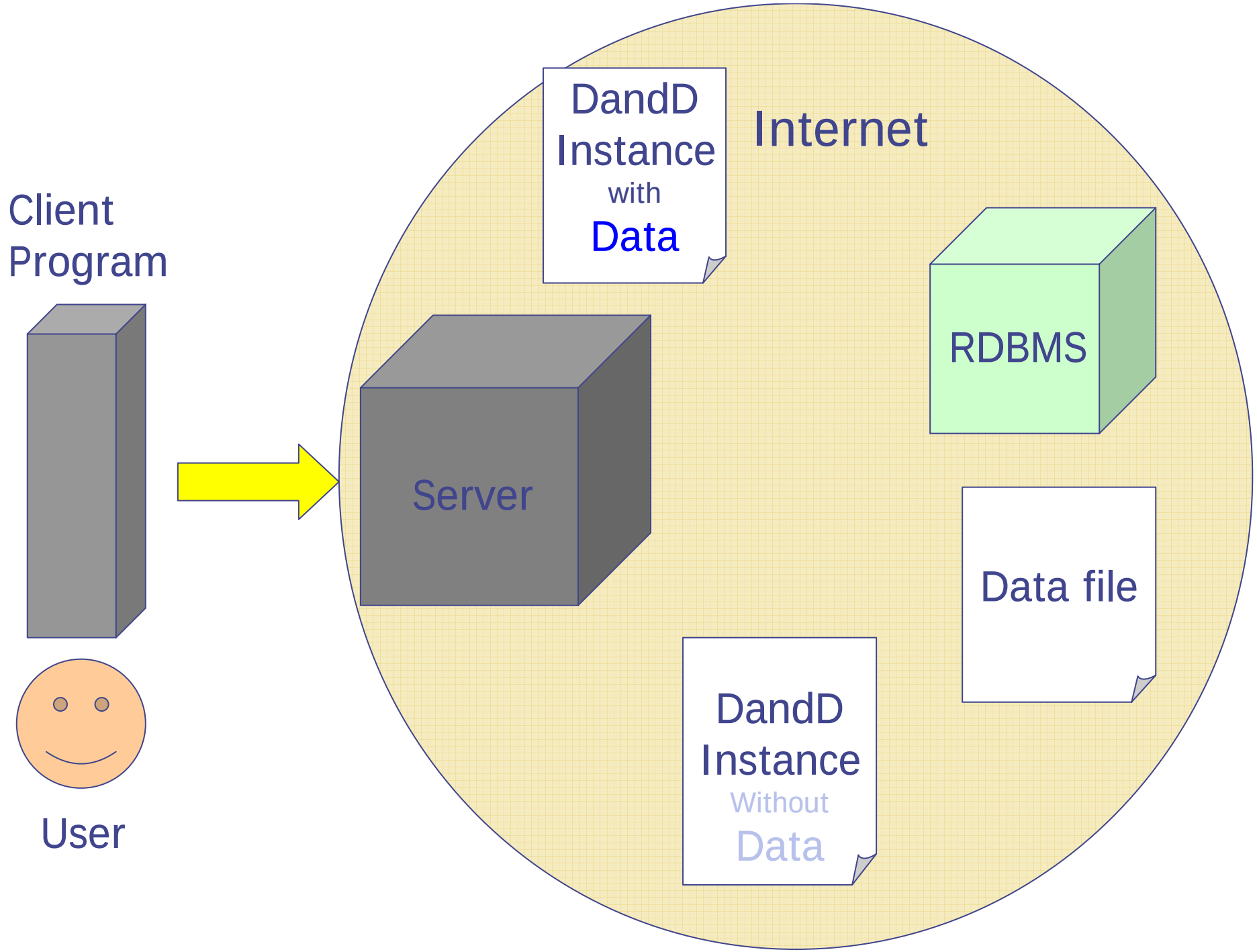
An example of DandD instance

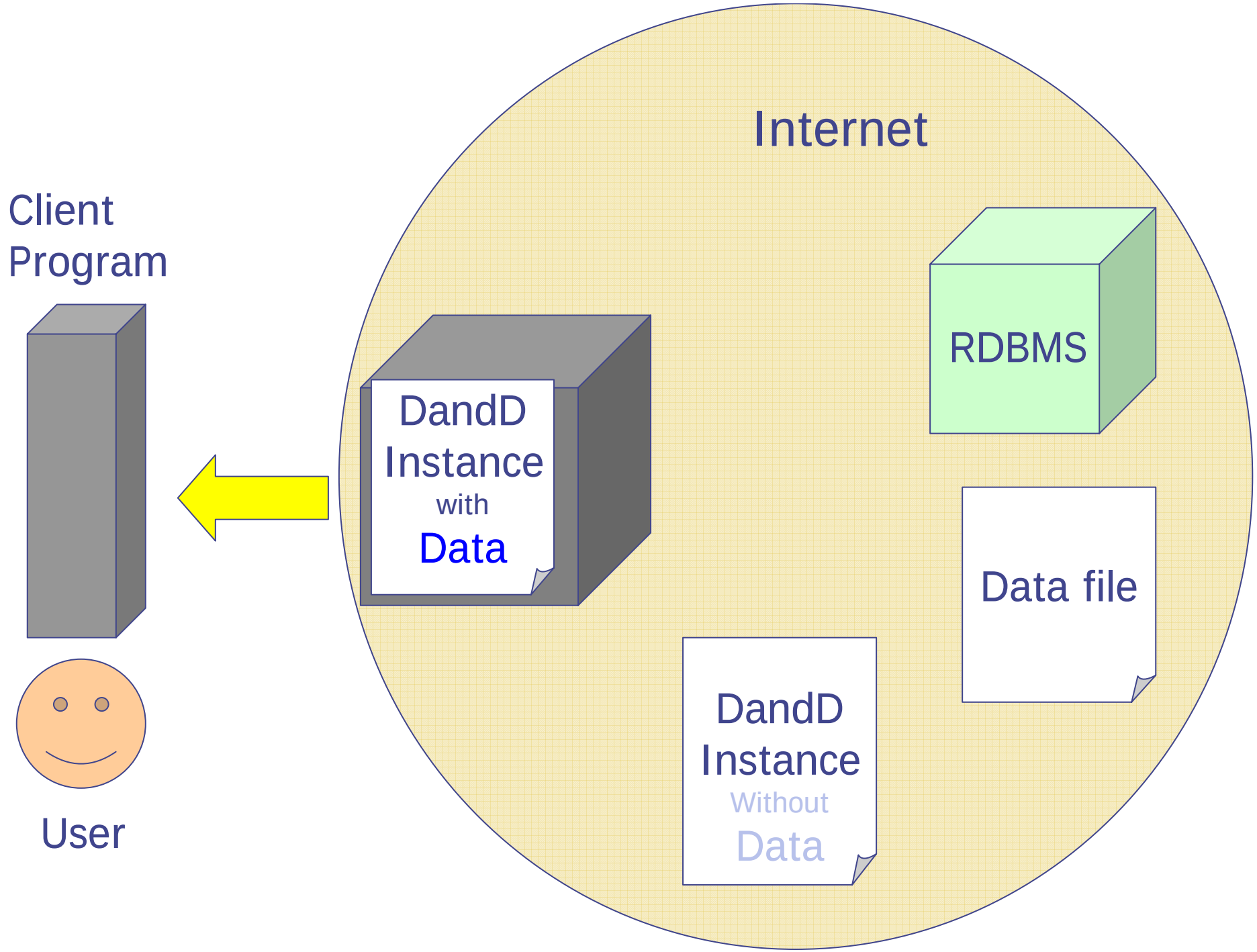
```
<Data>
<Relational Id="r1">
  <Value RefId="year" Systems="t1"/>
  <Value RefId="month" Systems="t1"/>
  <Value RefId ="day" Systems="t1"/>
    ...
 </Relational>
</Data>
<DataBody>
 <DataVector Id="year" >1975 2001 2003</DataVector>
 <DataVector Id="month" Code="c1 c2">1 2 3</DataVector>
 <DataVector Id="day" >9 3 10</DataVector>
  ...
</DataBody>
<Appendix>
   <Code Id="c1">"Oct." "Jan." "Sept."</Code>
   <Code Id="c2">"10月" "1 月" "9月" </Code>
</Appendix>
```

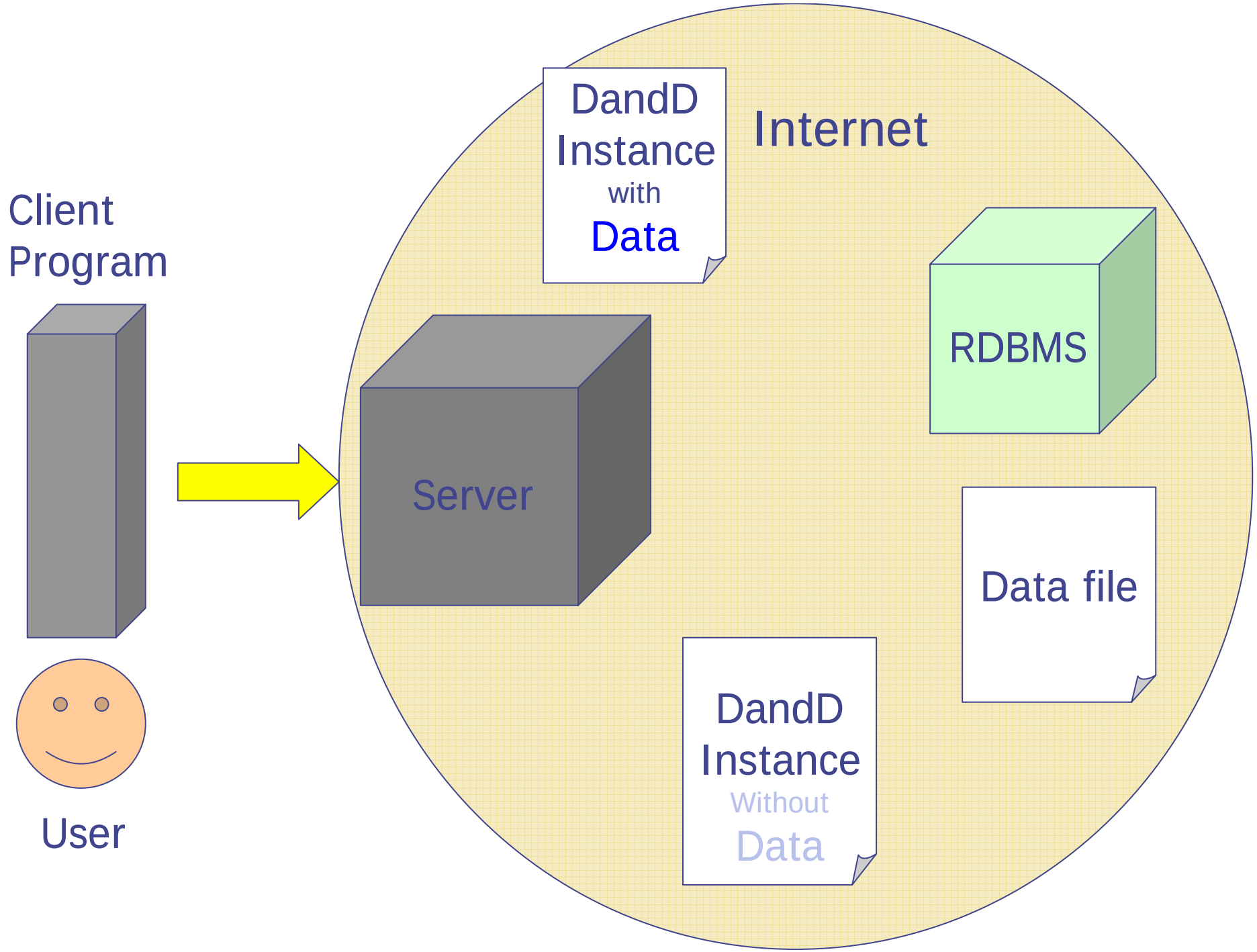
year	month	day	...
1975	10月	9	...
2001	1月	3	...
2003	9月	10	...

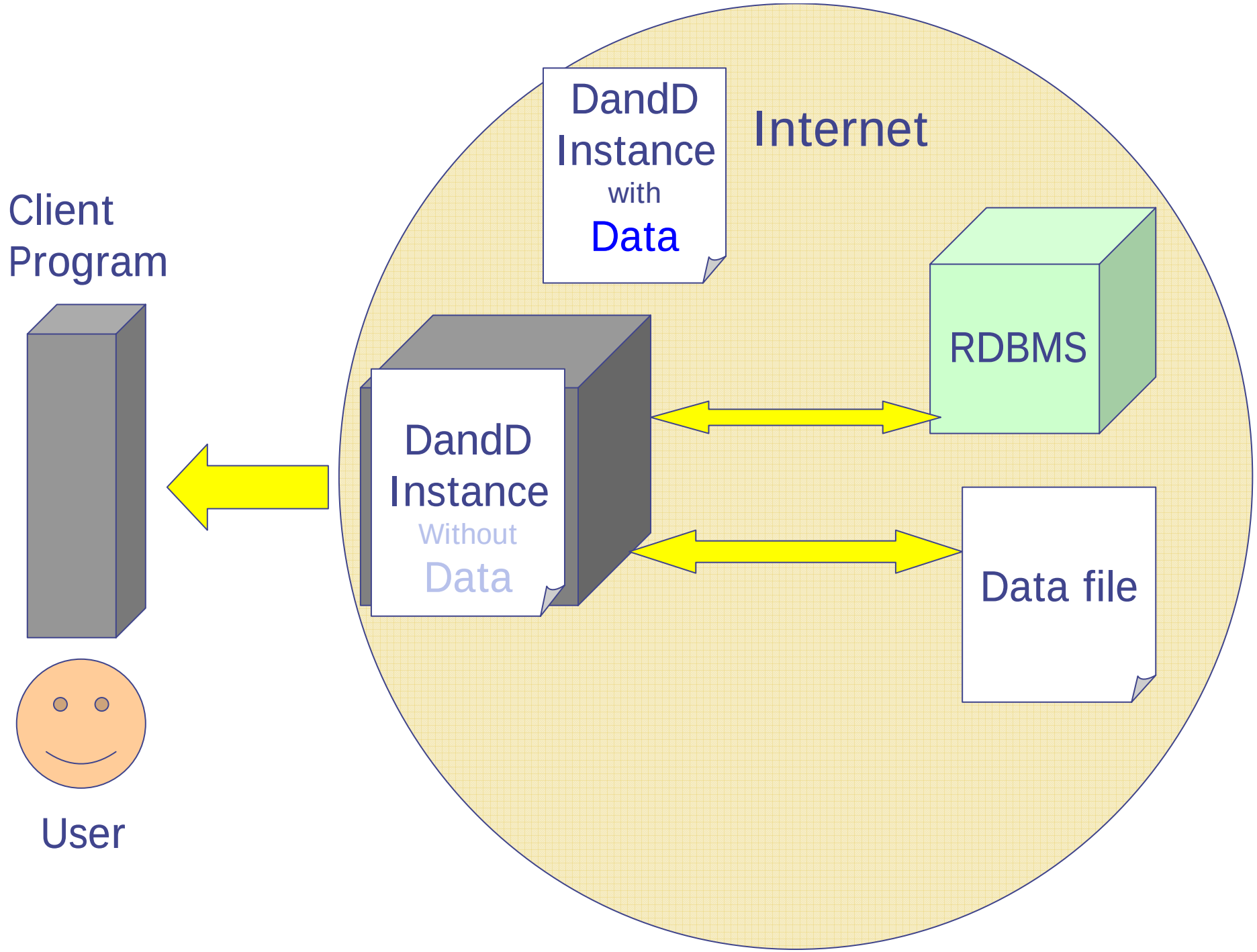
System definition

```
<Time Id="t1">
  <Year RefId="year"/>
  <Month  RefId="month"/>
   <Day RefId="day">
</Time>
```


Client
Program
User
Server
DandD
Instance
with
Data
Internet
RDBMS
Data file
DandD
Instance
Without
Data


Internet
Client
Program
DandD
Instance
with
Data
RDBMS
Data file
DandD
Instance
Without
Data
User


Client
Program
User
Server
DandD
Instance
with
Data
Internet
RDBMS
Data file
DandD
Instance
Without
Data


Client
Program
User
DandD
Instance
with
Data
Internet
DandD
Instance
Without
Data
RDBMS
Data file

InterDatabase (2001)

- Databases on the Internet
  - Scattered data on the network
  - Accessible via network
- Interface to databases (DandD instance)
  - Different database management systems
  - Different storage types
- Principle
  - Unit : DataVector
  - Integration : Relational

External DataVector

- Body
  - Null
- Attributes
  - Access
  - Protocol
  - PostProcessing

```
<DataVector Id="i1"
            Access="a1"
            Protocol="p1"
            PostProccesing="scan" />
```

Example of external DataVector

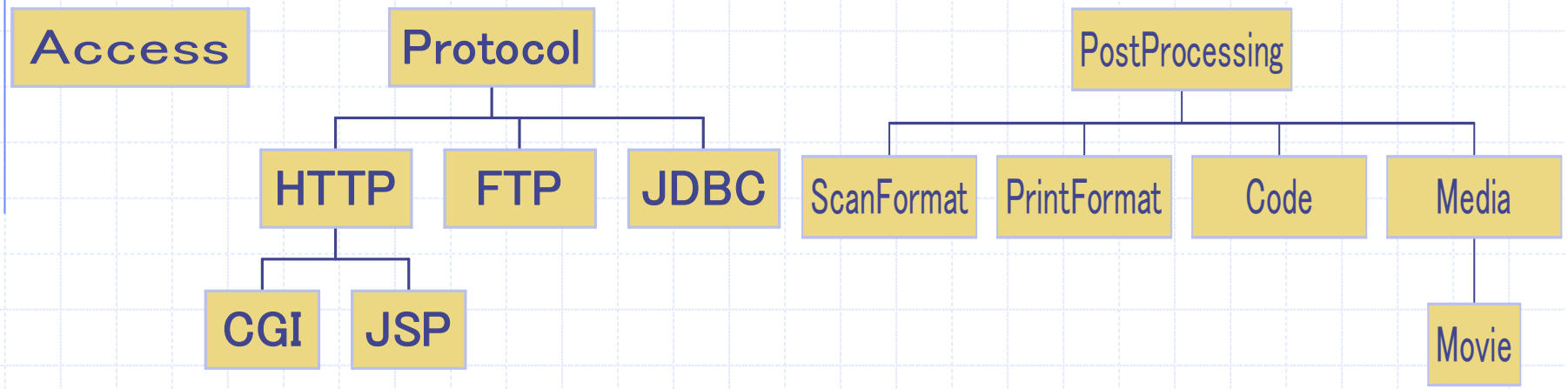
```
<DataVector Id="i1" Access="a1"
Protocol="p1"  PostProccesing="scan" />
<Appendix>
<Access  Id="a1" IP="131.113.65.1"
UserId="anonymous" />
<Protocol Id="p1" Physical="tcp">
  <FTP Suffix="/test.csv" Offset="3" Lines="28" />
</Protocol>
<ScanFormat Id="scan">
" %*s,%s,%*s"
</ScanFormat>
</Appendix>
```

ftp://131.113.65.1/test.csv

2003,12,11

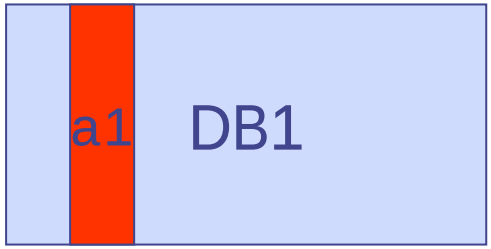
2003,12,12

Inheritance tree of Access, Protocol, and PostProcessing



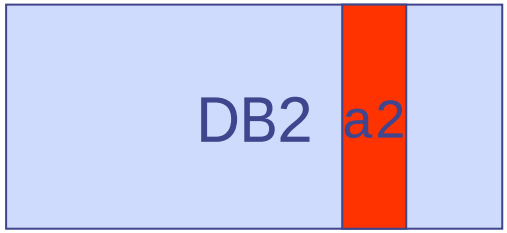
Integration of external DataVectors

```
<Relational Id="r1">
        <Value RefId="a1">
        <Value Refid="a2">
</Relational>
```

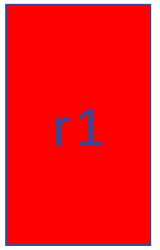


<DataVector id="a1" /> (exists in DB1)

<DataVector id="a2" /> (exists in DB2)



Scattered into two Databases



Virtual cbind in R

Integration of external DataVectors(2)

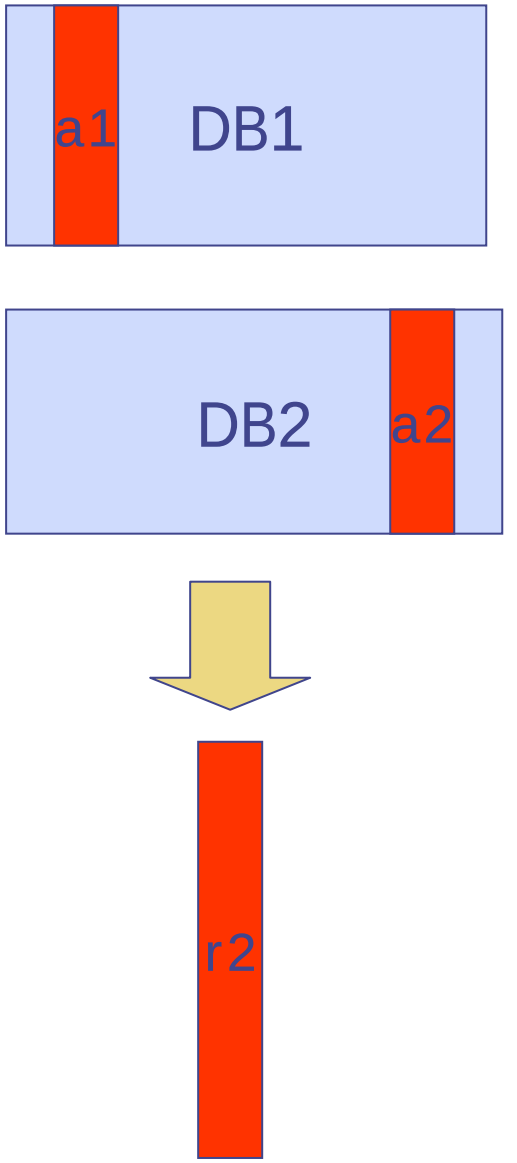
```
<Relational Id="r2">
        <Value RefId="a1 a2">
        .....
</Relational>
```

<DataVector id="a1" /> (exists in RDB1)

<DataVector id="a2" /> (exists in RDB2)

Integration of the attributes is necessary in this case !!

→ **virtual rbind in R**

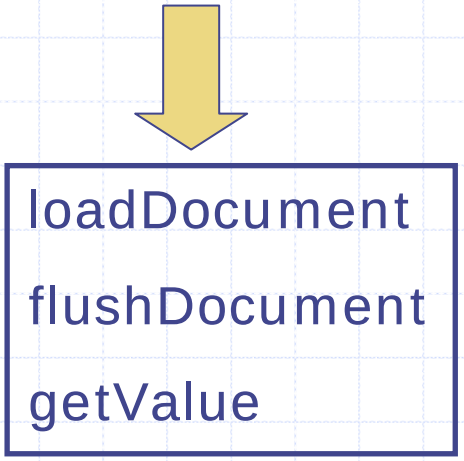


DandD Client Server System

- Advantage of client
  - Platform independent
    - Communication through socket
  - Easy programming
    - Query and Response
  - Mobility
    - cellular phone
    - PDA

DandDServer

- Recieve : DOM (Document Object Model) methods + Original methods
- Send : Flag+Size+String
- Implementation by Java
  - XML Parser
    - Xerces for Java2 (Apache)
  - Interpreter
    - Pnuts (Sun Micro Systems)

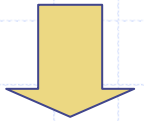


Client System

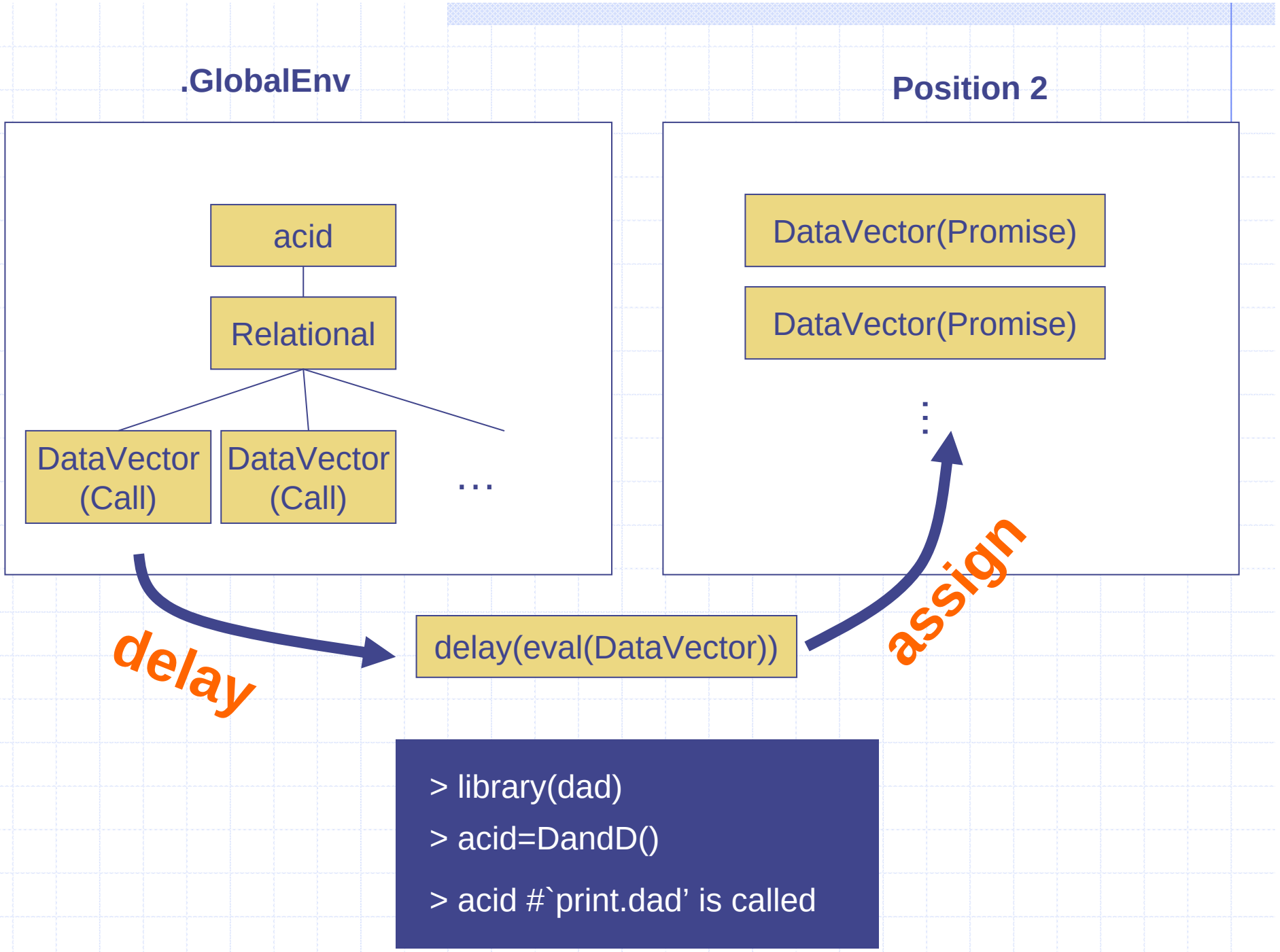
- Browsing
  - DandDBrowser
- Analysis and Modeling
  - DandDR
- Generation and Modification
  - DandD Instance Generation
  - Edit or update

DandDR

```
>library(dad)
>foo=DandD(“Acid.dad”) # return DandD Object
>foo
Acid Rain (Relational) attached to "AcidRain.dad:AcidRan"
[1] "SlftCcn"(Sulfate Cocentration)     "NtrtCnc"(Nitrate Concentration)
[3] "CntnsPr"(Continuous Precipitation)
>plot(foo) # visualization of DandD object
```



Promise Objects
by delay method

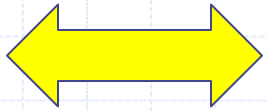

.GlobalEnv
Position 2
acid
Relational
DataVector (Call)
DataVector (Call)
...
DataVector(Promise)
DataVector(Promise)
delay
delay(eval(DataVector))
assign
> library(dad)
> acid=DandD()
> acid #`print.dad' is called

DandDR

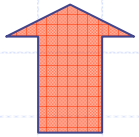
```
>library(dad)
>foo=DandD(“Acid.dad”) # return DandD Object
>foo
Acid Rain (Relational) attached to "AcidRain.dad:AcidRan"
[1] "SlftCcn"(Sulfate Cocentration)     "NtrtCnc"(Nitrate
Concentration)
[3] "CntnsPr"(Continuous Precipitation)
>plot(foo) # visualization of DandD object
```

Visualization

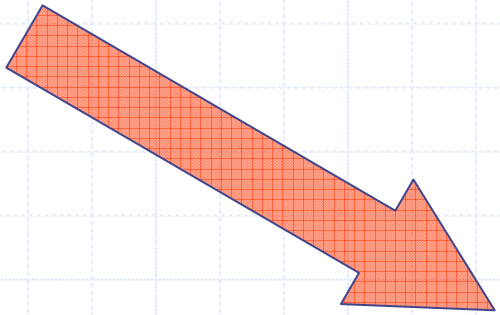
Data View



Visualization



DandDR



DandD Instance

Data organization

Various attributes

Background information

Visualization by DataType attribute of DataVector

- Conditional variable
  - Conditional
    - Unordered
    - Ordered
    - Equispaced
    - Binary
  - Auxiliary
    - Number
    - Sequence

→ **X axis**

- Nominal variable
  - Id

→ **Label**

- Resulting variable
  - Categorical
    - Category
    - OrderedCategory
    - Logical
  - Measurement
    - Score
    - Interval
    - NULL
  - Count
    - Count

→ **Y axis**

Homepage

- DandD Project Homepage
  - Examples of DandD instance
  - DTD
  - Software
- URL

http://www.stat.math.keio.ac.jp/DandDIII/