
# User-Driven Situational Service Mashups

**Guiling Wang**

**Research Center for Cloud Computing,**

**North China University of Technology,**

**Beijing, China**

**wangguiling@ict.ac.cn**

# Background

- **Situational Applications[1]/Situated Software[2]**

**“situational application** is "good enough" software created for a narrow group of users with a unique set of needs. ... As the requirements of a small team using the application change, the situational application often also continues to evolve to accommodate these changes. ”

- **For example, situational data integration application:**
  - **Data sources can not be enumerated exhaustively**
  - **Requirements are in great variety**

Mashup is a new application development method that allows non-professional users to build applications by combining functionalities offered by more than one source to deal with situational and ad-hoc problems.

Software, 2008, 25: 30–36

# IT Developer Driven ->User Driven

- **IT Developer Driven**
  - It is difficult for the IT developer to build a system to satisfy the diverse, transient user requirements which can not be determined in advance
- **User Driven**
  - To support end-users, business users and casual programmers using short development life cycles to solve emergent or ad-hoc problems, and are updated frequently as needs evolve
  - End-user programming promises a development fashion for creating situational applications

# A Motivating Application Scenario

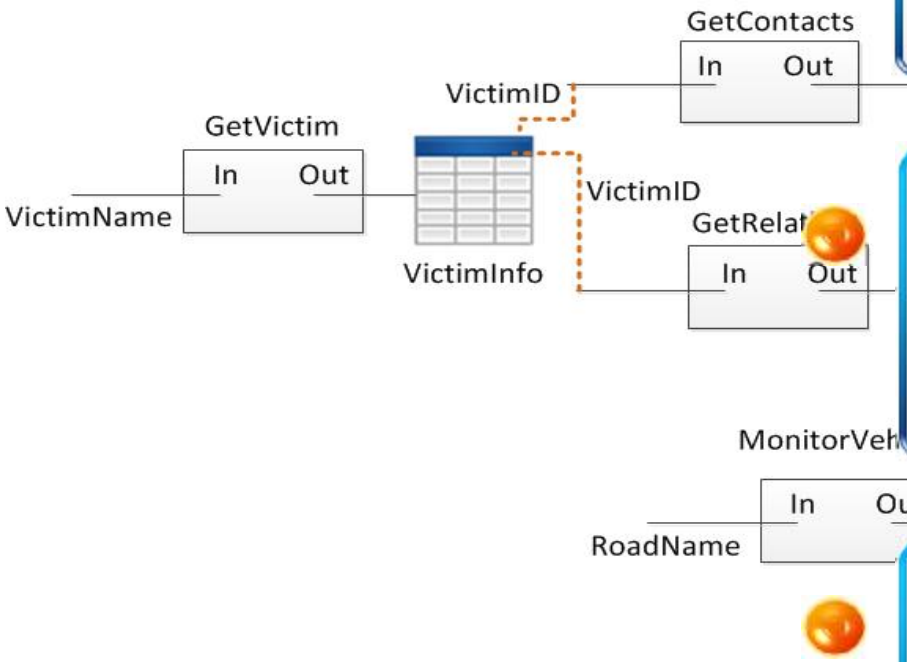
- **Who is the Criminal Suspect?**

# Context Information as the Clues

- **Context information about the crime : location** of the crime/crime **items, fingerprint,** monitoring video, **mobile phone signal**


  - **location of the crime is far from the place where the crime items are found=>**
  - **the criminal must have transported the crime items=>**
  - **Compare vehicles passed through the road and vehicles of the victim's social relationships**

# Crime Analysis Mashu



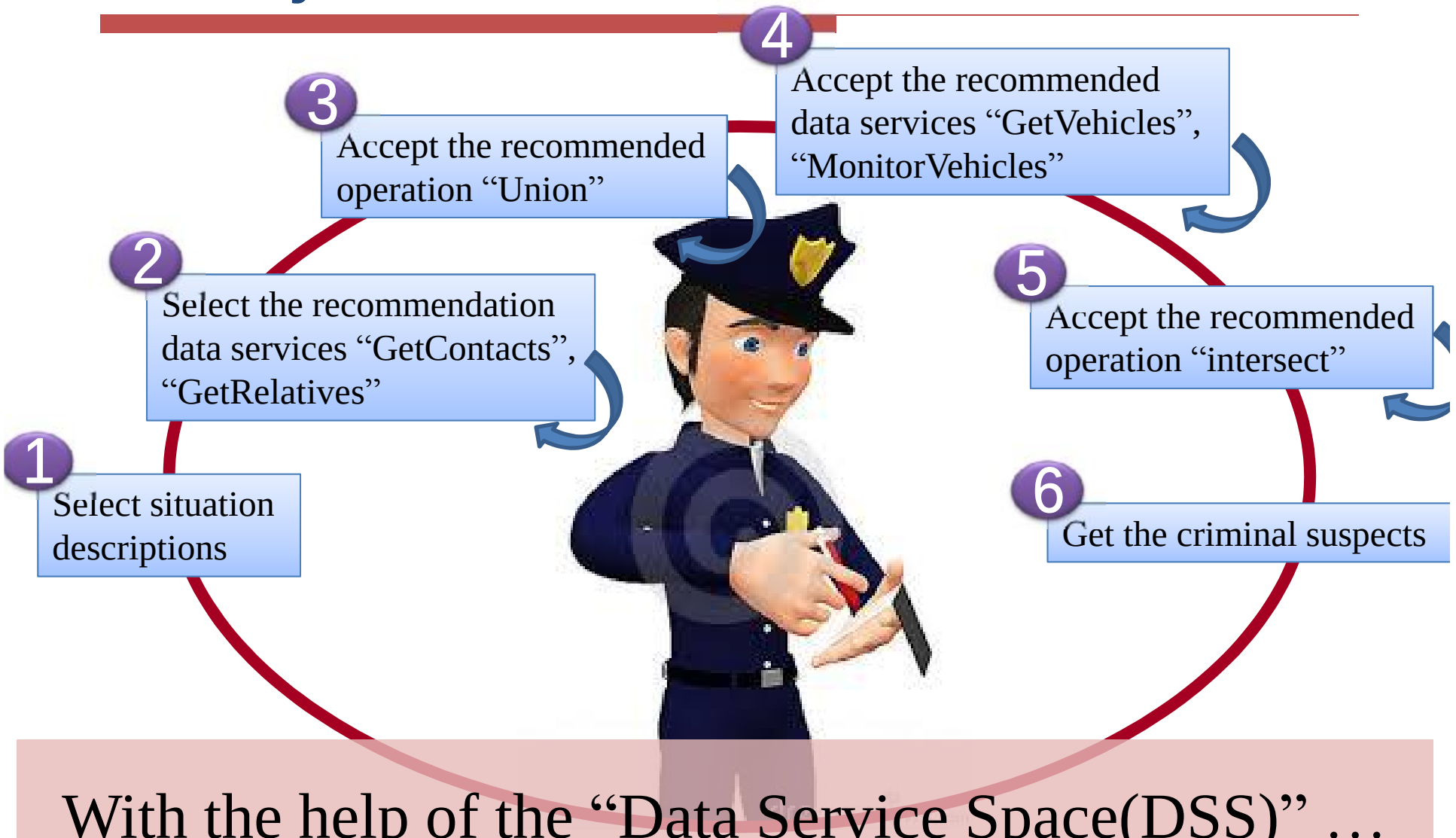

There are so many potential data sources, how to select from them?

It is difficult to process and combine the data sources for non-professional programmers. Which step should they take next?

For the different criminal cases, the situational information is deemed different. How to adapt to the current situation?

- Get the social relationship of the v
  recent contacts
- Get the vehicle information of the victim's social relationship
- Get the information of the vehicles passing through the road between crime scene and the place of crime items
- Get the suspect list by comparing the vehicle information

# Previously Tedious/Difficult Work -> Present…




With the help of the “Data Service Space(DSS)” …

# Some Research Issues:

- **It will be very helpful if we *provide some recommendations on what to do next*. How to do recommendations?**
  - **How to describe the situation? Given the description of the situation, *recommend a set of data services related with the situation.* Given partial mashups, *recommend the operations/data services ?***
  - **Where dose the data service come from? It is complex to access various data sources, including HTML pages, databases, APIs, etc. How to *provide a uniform abstraction of data sources* ?**
  - **It is complex for end users to express the service composition plan of situational applications. How to *provide a visualized programming environment for end users to mashup the data services*?**

# Research Agenda

DeCloud DataServiceSpace

- **Support end users or business users to develop the situational application, and aid them by suggesting helpful recommendations.**


- **http://113.11.194.86/DataServiceSpace/index.jsp**

# Data Services

- Nested table model(intuitive and closer to the real world)
- Semi/Automatically transformation of HTML/JSON/XML
- Both input and each column of the output nested table is associated with a set of tags

**Schema:=**
CallRecords {id, name, identityNo, record{$phone_1$, $phone_2$, startTime, duration}}
**Tags:**
{name<victimName>,
identityNo<identityNo,victimeID>,
phone<phone,mobile phone,tel>,
startTime<start time>,
duration<call duration>}
**Instances:**
{ <001, San Zhang, 11011xxxxxx, {
<130xxxxxx, 131xxxxxx, 2013-5-1 16:03, 15min>,
<130xxxxxx, 133xxxxxx, 2013-5-2 17:01, 5min>,
.....
} >,
<002, Si Li, 22022xxxxxx, {
<133xxxxxx, 136xxxxxx, 2013-5-1 10:03, 12min>,
<133xxxxxx, 139xxxxxx, 2013-5-3 13:01, 7min>,
.....
} >
...
}

(a) Tagged Nested Relations for the Output of the queryCallRecords Service

| CallRecords | | | | | | |
|---|---|---|---|---|---|---|
| id | name | identityNo | record | | | |
| | | | phoneNo1 | phoneNo2 | startTime | duration |
| 001 | San Zhang | 11011xxxxxx | 130xxxxxx | 131xxxxxx | 2013-5-1 16:03 | 15min |
| | | | 130xxxxxx | 133xxxxxx | 2013-5-2 17:01 | 5min |
| 002 | Si Li | 22022xxxxxx | 133xxxxxx | 136xxxxxx | 2013-5-1 10:03 | 12min |
| | | | 133xxxxxx | 139xxxxxx | 2013-5-3 13:01 | 7min |
| ... | ... | ... | ... | ... | ... | ... |

Schema
Instances

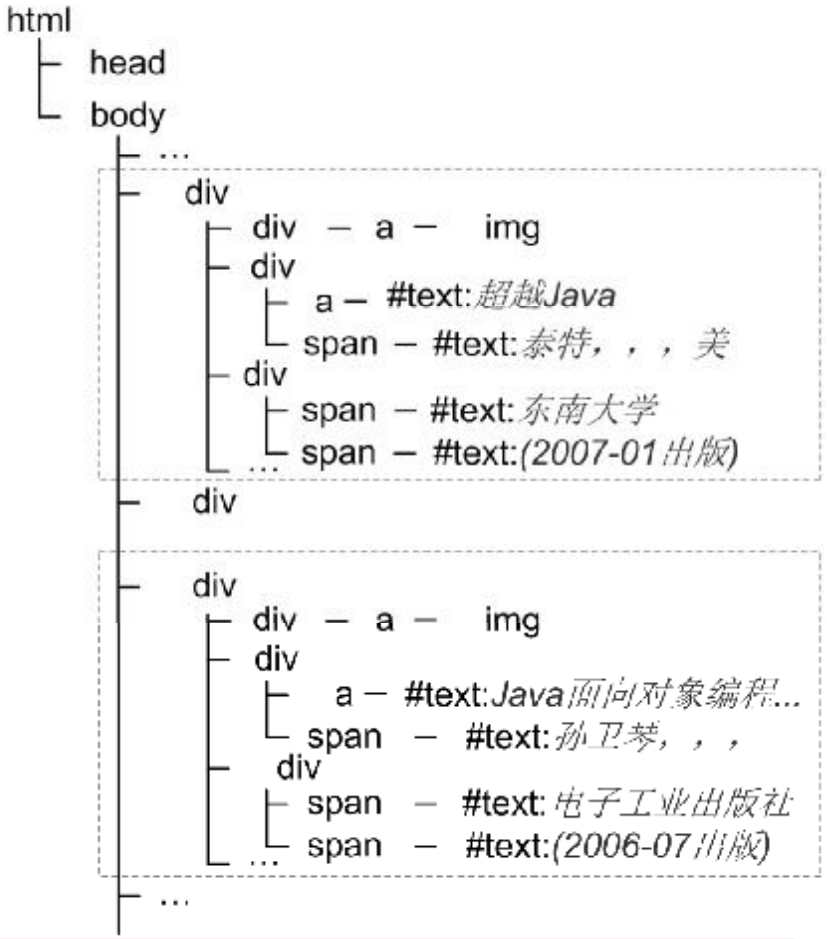
Recommendation
GetRelatives
GetVehicles

(b) Tagged Nested Table for the Output of the queryCallRecords Service

# Interactive Data Service Generation Method for HTML Pages: GRUB

- **Similarity assumption, supervised learning**


| Book | | |
|---|---|---|
| *title* | *author* | *price* |
| *超越Java* | *泰特* | *21.20* |
| *Java面向对象编程...* | *孙卫琴* | *49.30* |
| ... | ... | ... |



# Evaluation

Ji Guang, Wang Guiling, Han Yanbo. Creating customized data services from web pages. High Technology Letters, Vol.19 No.2, pp.203-207, 2013

- **Effectiveness** http://113.11.194.86/DataServiceSpace/index/core/create_data_service_clipper.jsp
  - ***Precision=98.20%*** **,** ***Recall=98.95%*** **for record block,** ***Precision=Recall=91.34%*** **for record attribute. GRUB has basically as effective as typical related works**
- **Usability**
  - **Users don't need to master programming knowledge, such as HTML language and regular expression**
  - **Generate a data service within two minutes**
- **Flexibility**
  - **supports both record-level and page-level information extraction while no related works support both of them up to now.**

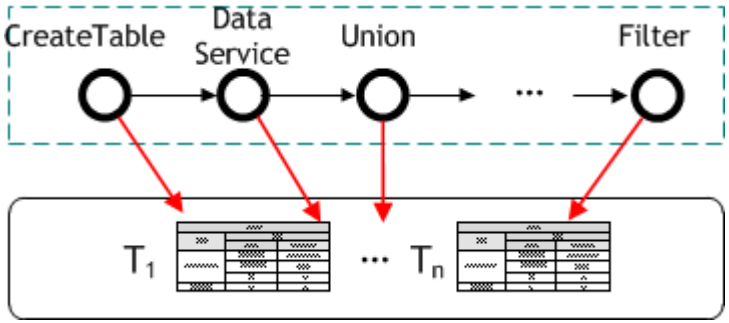
# How to provide an easy-to-use data service composition environment?

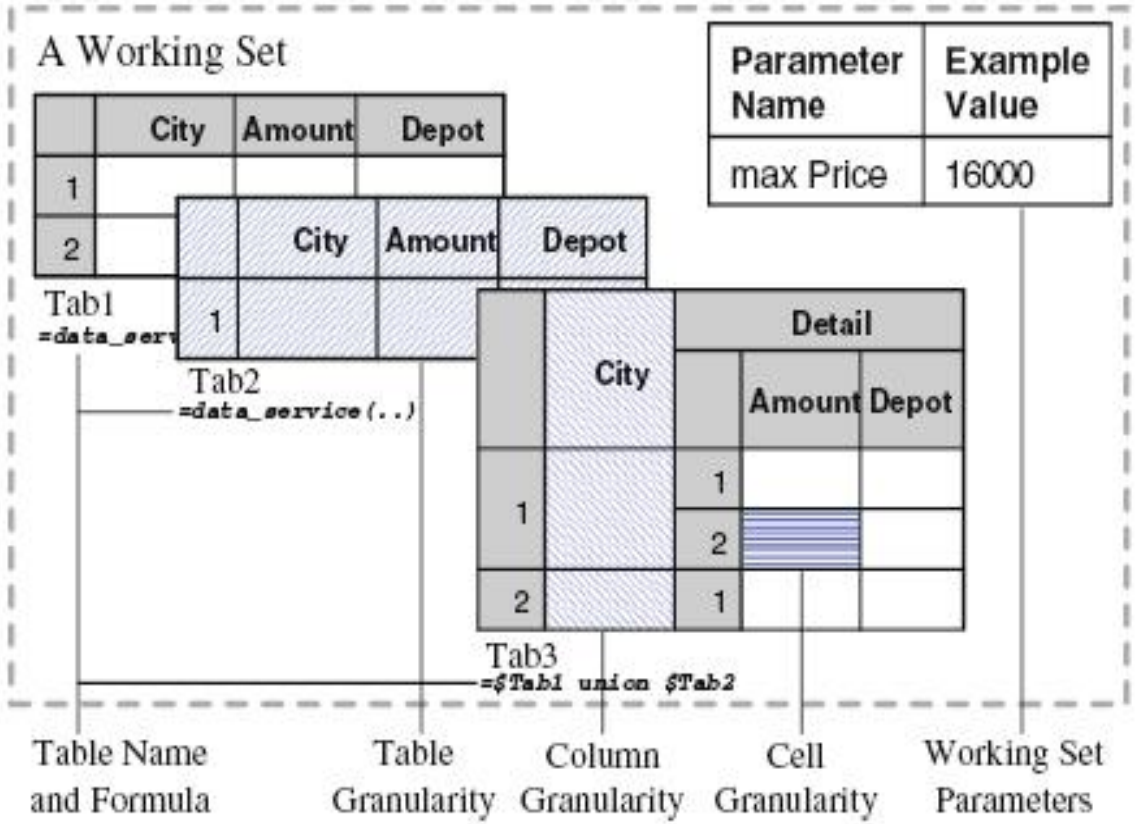
- **The traditional service composition environment is for IT users, not for business users or end users**
- **There are no adequate data service composition environment for business users**
  - **Data flow programming pattern: users often feel awkward in relating the input with output**
  - **Spreadsheet programming pattern: two-dimentional spreadsheet can not process and present the complex data structure**
  - **Tree-like programming pattern: hard to understand the relationship between tree nodes**

A new visualized data service operation environment is needed.
It requires agility and rich expressive power
at the same time

# Graphical Data Service Composition Environment Using Nested Tables

- Programming by Example
- Multi-granularity user operations: Table, Column, Cell

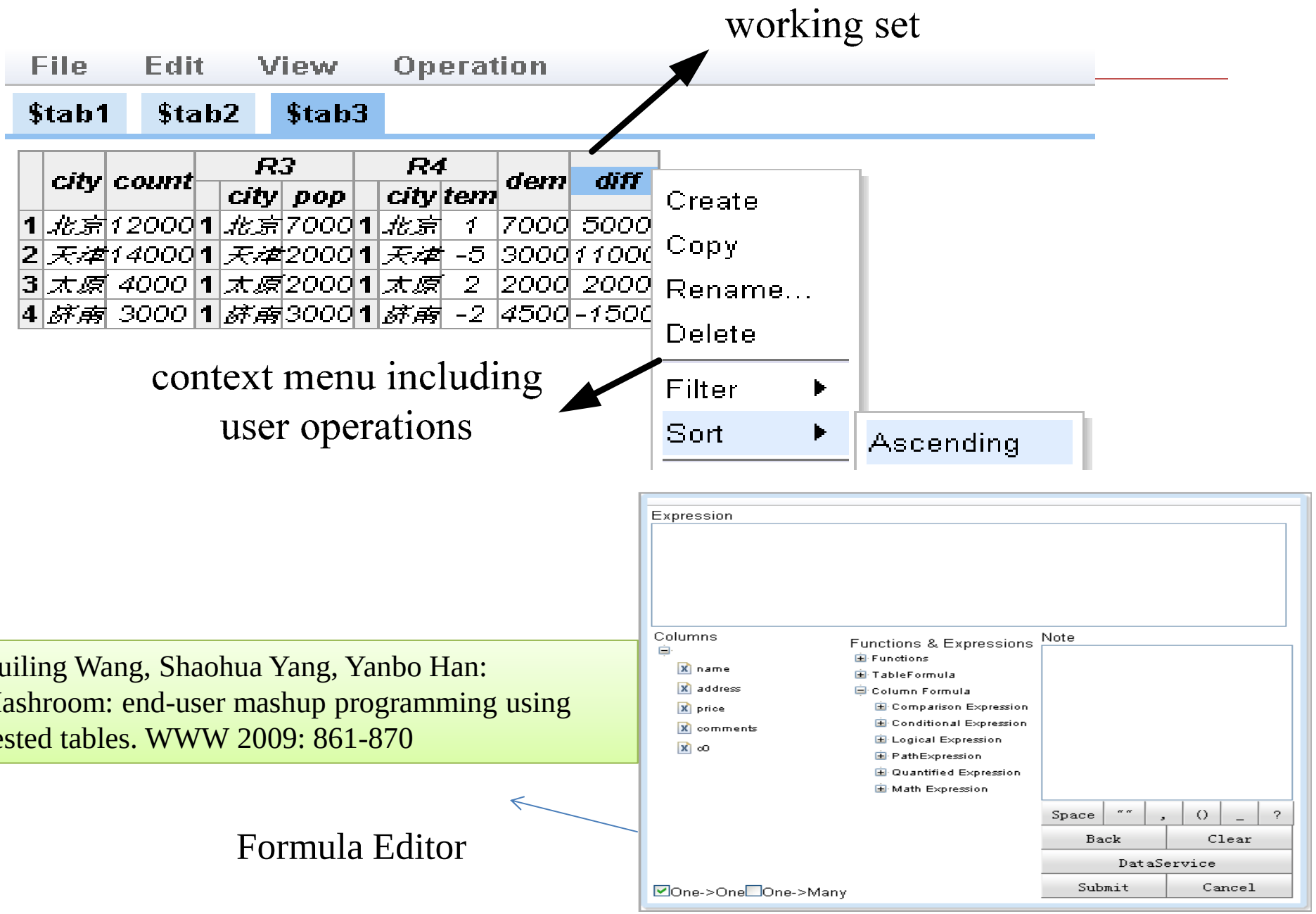


# Nested Table Algebra

- **The set of nested tables *S* *together with the d* , ρ, μ n β, γ**

| Operation Classification | Data Operation | | Corresponding basic operation for tranditional nested relation |
|---|---|---|---|
| Data Accession & Normalization | Import ( ξ ) | | - |
| Data Transform | unary operation | AddColumn ( α ) | - |
| | | UpdateColumn ( β ) | - |
| | | DeleteColumn ( γ ) | project |
| | | Rename ( ρ ) | rename |
| | | Unnest ( μ ) | unnest |
| | | Nest ( ν ) | nest |
| | | Copy (c) | - |
| | | Sort ( ο ) | - |
| | | Truncate ( κ ) | - |
| | | Filter ( δ ) | select |
| | | LinkService ( τ ) | - |
| | | MergeTuples ( λ ) | - |
| | Binary operations | Cartesian Product (×) | Cartesian product |
| | | Join (⋈) | - |
| | | Union (∪) | union |
| | | Difference ( − ) | intersect |

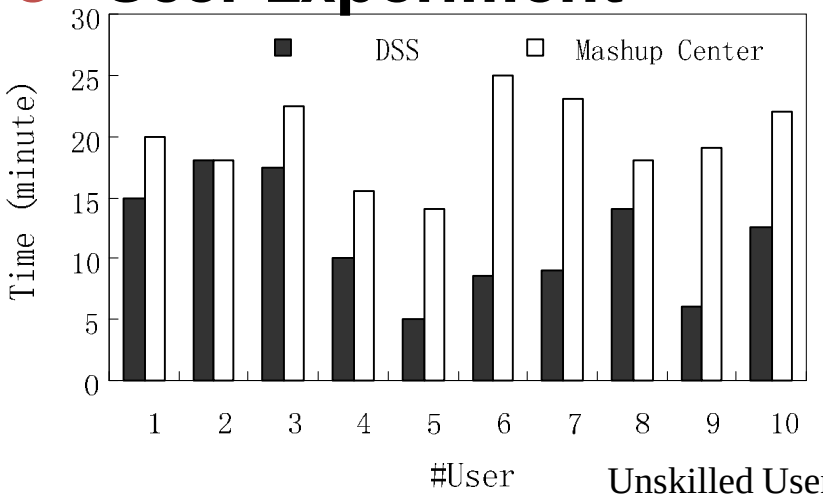


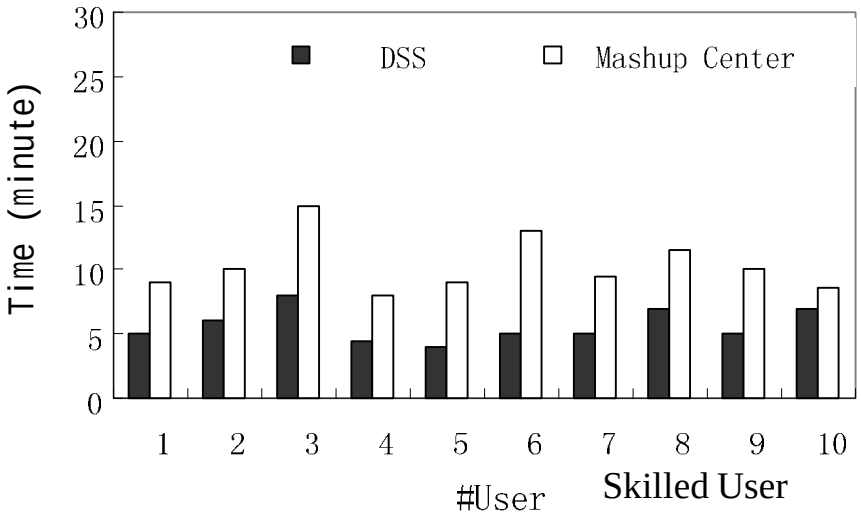

Guiling Wang, Shaohua Yang, Yanbo Han: Mashroom: end-user mashup programming using nested tables. WWW 2009: 861-870

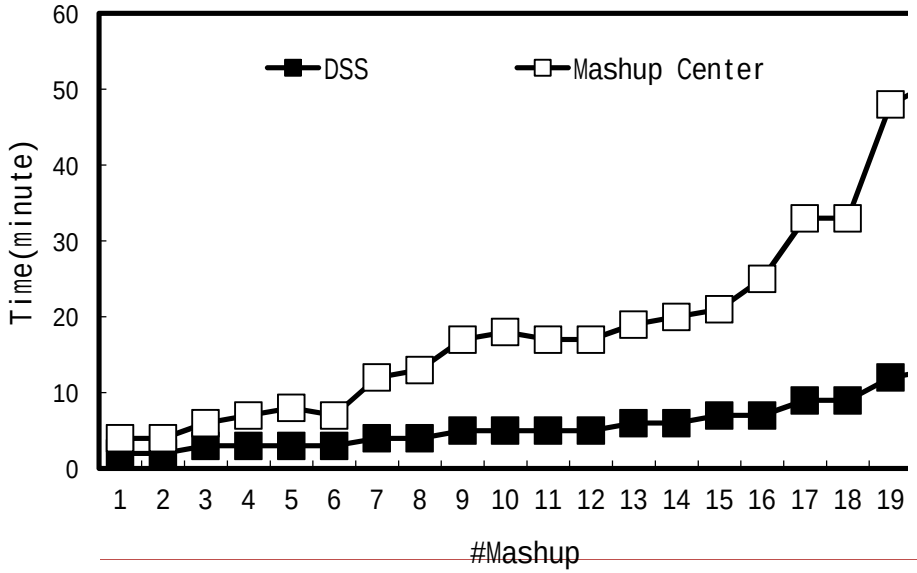
# Evaluation

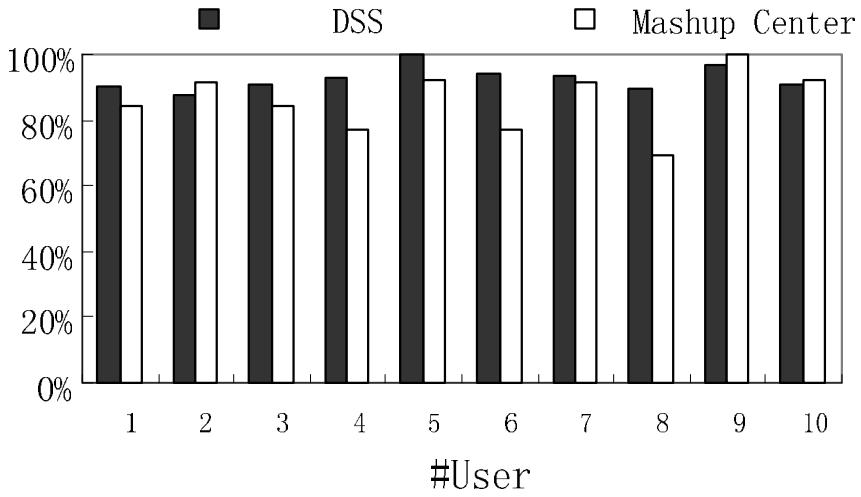
Yanbo Han, Guiling Wang, Guang Ji, Peng Zhang: Situational data integration with data services and nested table. Service Oriented Computing and Applications 7(2): 129-150 (2013)

## o User Experiment

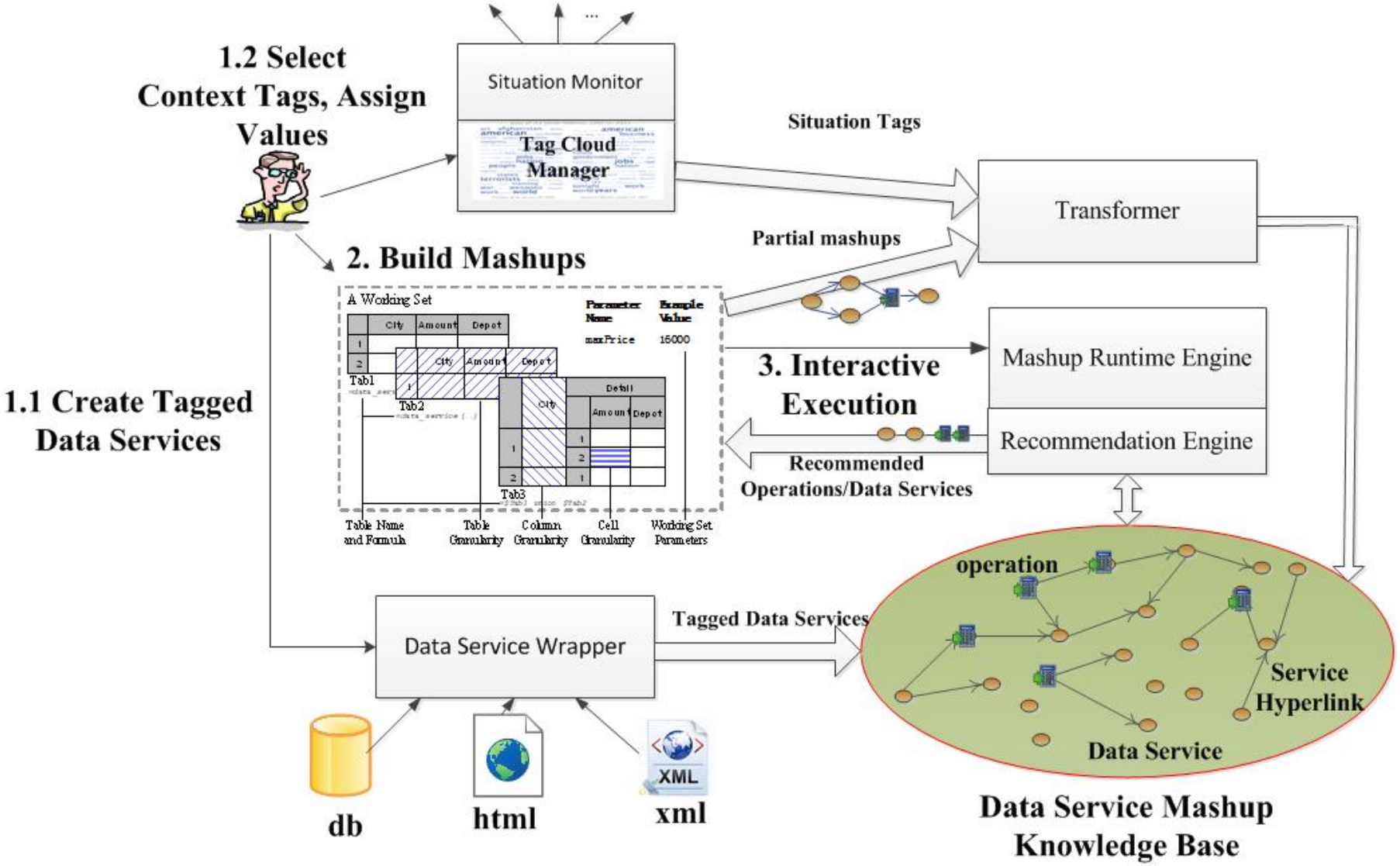


# Our On-going Work : an Initial Approach to Recommendation

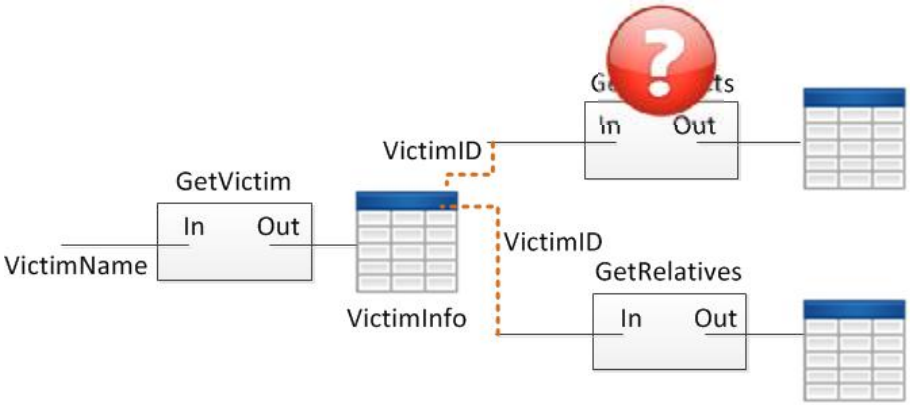


# Example Situations for our Data Services

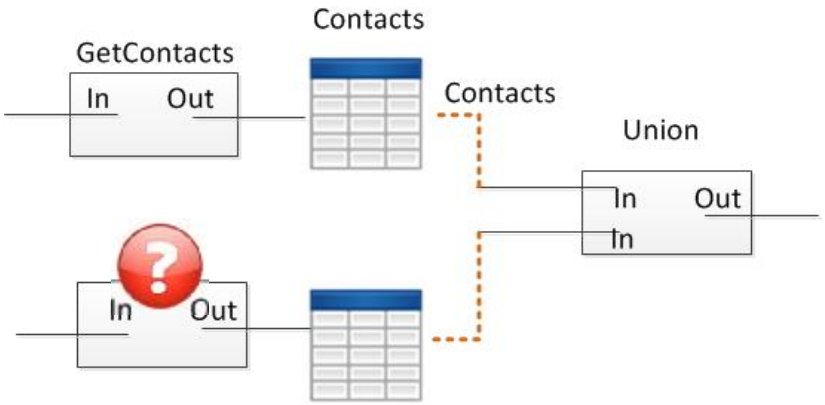
- **Find previous top-*k* situations similar to the current situation. Then calculate the recommended data services emerged in these situations.**
- **Model context as a set of attributes**
  - ***e.g., C={crime location, crime items location, … }***
- **A situation can be represented as a vector. Each element in the vector is value of context attribute**
  - ***e.g., s=<Chaoyang Beijing, Haidian Beijing, …>***
- **Given two situations $s=<c_1,c_2,\ldots,c_n>$ and $s'=<c_{1'},c_{2'},\ldots,c_{n'}>$, the similarity between s and s′ (represented as sim(s,s')) is calculated from:**
  - $sim(s,s') = \sum_{i=1}^{n} w_i * sim_i(c_i, c_i')$

# Recommend Target/Source Operator based on Statistics of Mashup History Logs

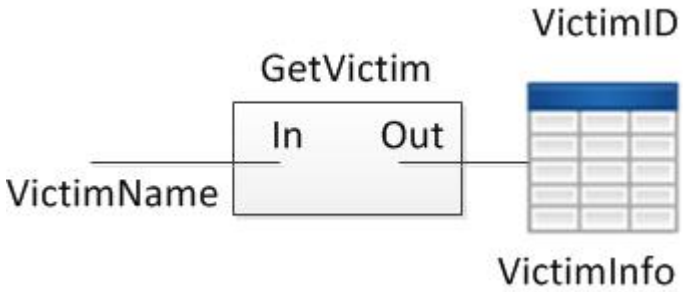
- **Recommend the target operator**




- **Recommend the source operator**



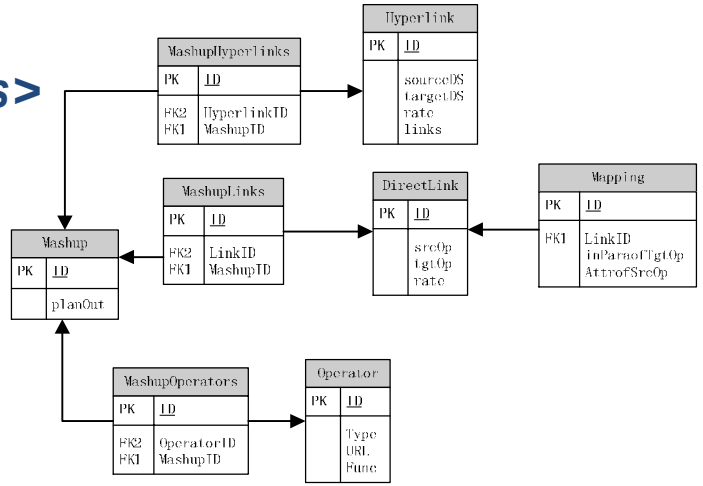

- **Recommend the related data services**




# Recommendation based on Statistics of Mashup History Logs

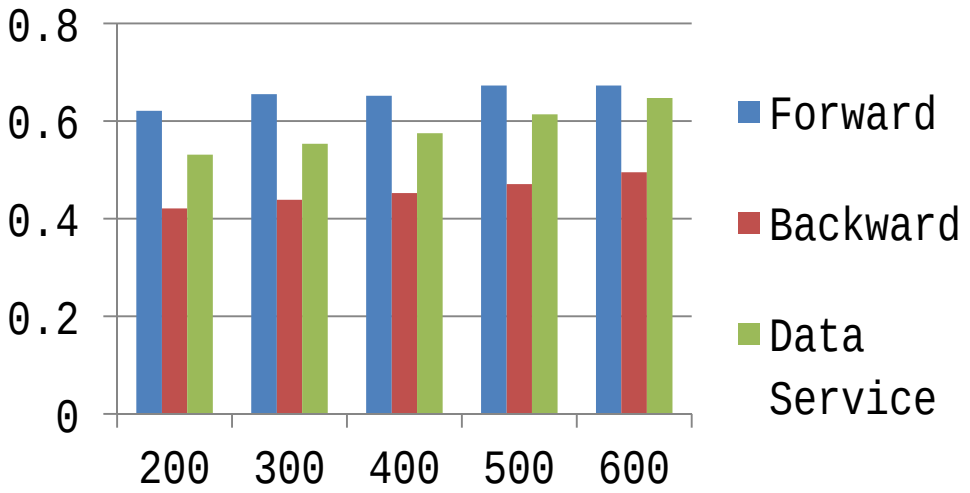
- ***MashupPlan=<Ops,Connectors,PlanIn,PlanOut, Situation>***
- ***connector = <Link, Mappings>***
  - ***Link=<$Op_{src}$,$Op_{tgt}$>,Link can be represented as $Op_{src}\rightarrow Op_{tgt}$ for simplicity,***
  - ***$Op_{src}$ is the source operator of the connector,***
  - ***$Op_{tgt}$ is the target operator of the connector,***
  - ***Mappings = {$m_l$| $m_l$ =<$OpIn_l$, $A_q$>, $A_q \in T_{src}$}.***
- **If two data services can be connected by direct or indirect connectors, we call there exists a hyperlink between these two data services. Connectors is the set of connectors between *$DS_{src}$* and *$DS_{tgt}$*,**
  - ***hyperlink = <$DS_{src}$, $DS_{tgt}$, connectors>***

$$Rate(a \rightarrow b) = \begin{cases} \sum_{o_i \in Ops_{DS}} Sim_{a,o_i} \cdot C_{o_i \rightarrow b}, a \in Ops_{DS} \\ C_{a \rightarrow b}, a \in Ops_{operation} \end{cases}$$



# Initial Experiment

- **Data set**
  - **Mashup history logs are from Yahoo! Pipes, totally 620 pipes(after pre-processing, 4857 operator connectors)**
  - **600 pipes as sample data set , 20 as the test data set**
- **select *k*=5 and select 200, 300,..., and 600 pipes**




# Acknowledgement

**Research Center for Cloud Computing, North China University of Technology**
**Prof. Yanbo Han and other colleagues**
**Graduated PH.D student: Dr. Shaohua Yang, Dr. Guang Ji**
**Master student: Sai Zhang, Bo Cao, Meizhen Geng**

# Thanks for your attention !

**Guiling Wang**

**Email : wangguiling@ict.ac.cn ,**

**wangguiling@ncut.edu.cn**