IPIT Collaboration –

Large-Scale Data-Driven Testing and Search-Based Optimization

Xiaoying Bai

Department of Computer Science and Technology

Tsinghua University

June, 2018

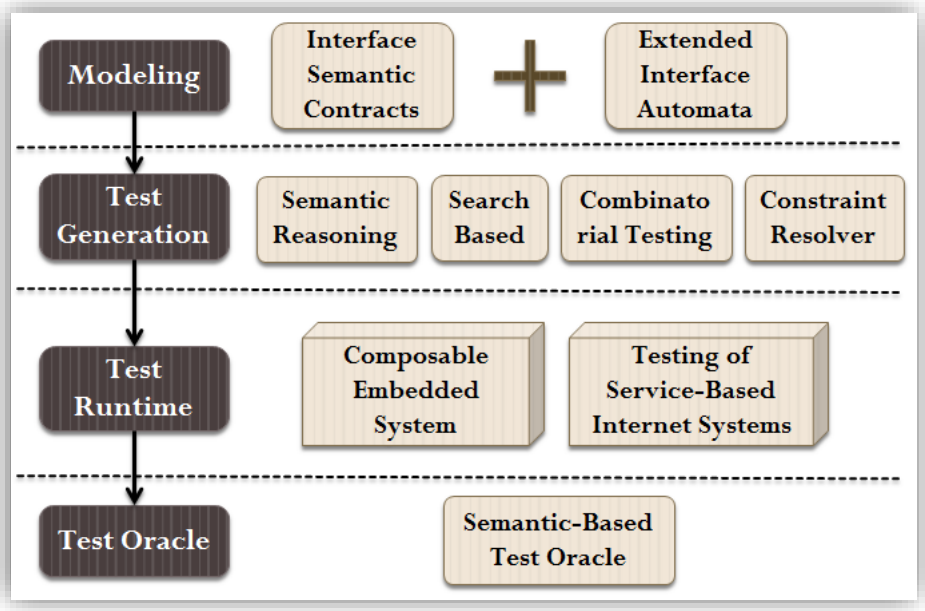
About Me: Research

Testing-in-the-Large

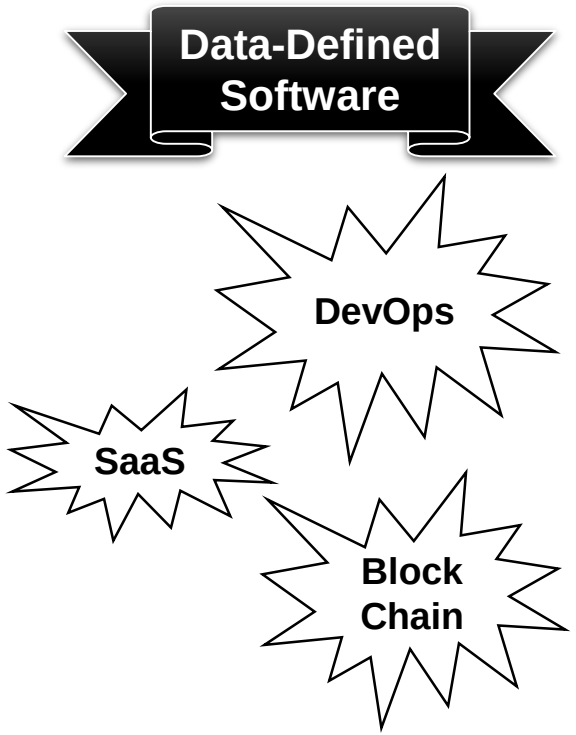
Ph.D thesis (1998-2001):
End-to-End Integration Testing:
A Thin-Thread Based Approach

2002-2008

Model-Driven Test Automation



Data-Driven Test Intelligence

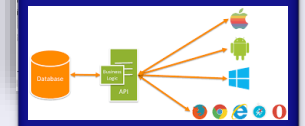


Stage 1 → **Stage 2** → **Stage 3**

Data-Driven Testing in the Paradigm Shift

Document

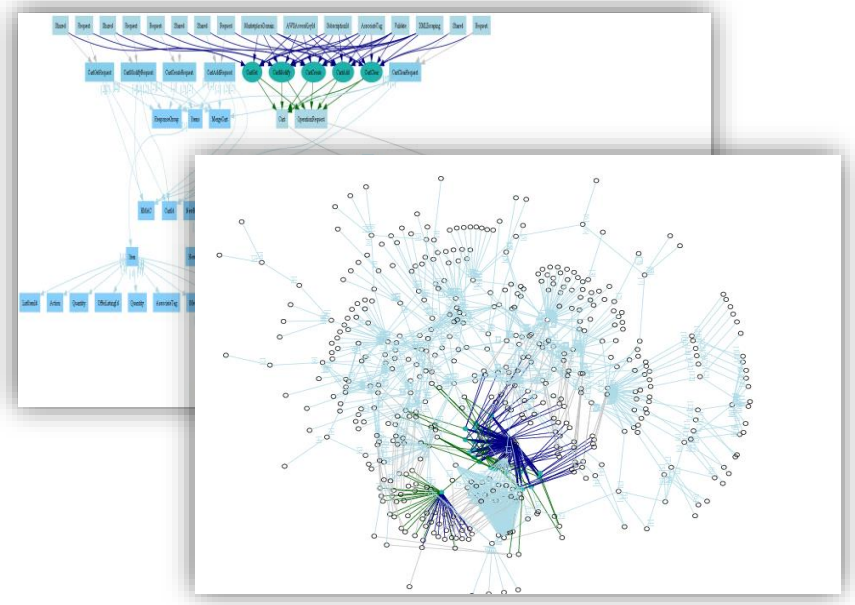
Web API



Program

服务化
可组合
软件系统

Data



Test Data Generation

How to achieve “data coverage” in terms of volume and variety?

How to effectively detect “data-sensitive” defects?

Some Observations

(Web) API is by nature the executable requirements of software components.

Test Generation

接口名称	输入参数	输出	异常信息	接口说明
激活会员 void activeMember (String memberNo)	memberNo: 会员编号(必填) **Domain Concepts**	无	MemberNotExistsException 会员不存在、待激活的手机或邮箱不存在 MemberBusinessException 会员状态异常(非注册中)	激活注册会员
绑定邮箱 void bindEmail (String memberNo, String email, String loginPassword) **Machine interpretable**	memberNo: 会员编号(必填) **Constraints** loginPassword: 登录密码 (当会员登录密码为空时此参数有效，场景：联合登录会员绑定邮箱时密码为空)	无	MemberExistsException 邮箱已存在异常 MemberBindException 邮箱绑定异常 MemberNotExistsException 会员不存在异常 MemberBusinessException 会员状态异常(正常、业务冻结可绑定邮箱) **Domain Knowledge**	如果是修改绑定邮箱，则原有的邮箱失效。原有的邮箱可重新用来注册。

Intelli-gence

Scale

Semantic Modeling

Optimized Search

Test Generation Based on Interface Semantic Contract

(Web) API is by nature the executable requirements of software components.

Test Generation

接口名称	输入参数	输出	异常信息	接口说明
激活会员 void activeMember (String memberNo)	memberNo: 会员编号(必填)	无	MemberNotExistsException 会员不存在、待激活的手机或邮箱不存在 MemberBusinessException 会员状态异常(非注册中)	激活注册会员
绑定邮箱 void bindEmail (String memberNo, String email, String loginPassword)	memberNo: 会员编号(必填) loginPassword: 登录密码 (当会员登录密码为空时此参数有效，场景：联合登录会员绑定邮箱时密码为空)	无	MemberExistsException 邮箱已存在异常 MemberBindException 邮箱绑定异常 MemberNotExistsException 会员不存在异常 MemberBusinessException 会员状态异常(正常、业务冻结可绑定邮箱)	如果是修改绑定邮箱，则原有的邮箱失效。原有的邮箱可重新用来注册。

Domain Concepts

Constraints

Machine interpretable

Domain Knowledge

Intelli-gence

Scale

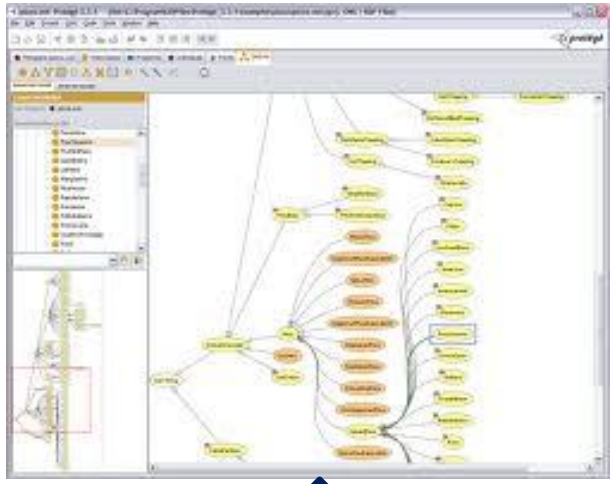
Semantic Modeling

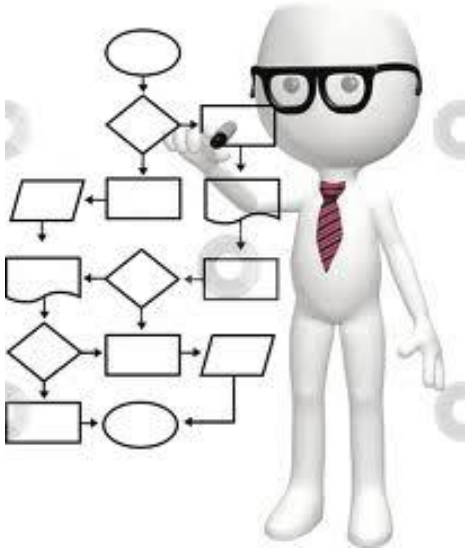
Optimized Search

From Implementation to Interface

```
@WebService
public class AddNumberImpl {
        @WebMethod (action = "…")
        publi
}
```

```
<WSDL: portType name="....">
    <WSDL: operation name = "...">
            <WSDL: input message ="..."/>
            <WSDL: output message ="..."/>
........
```





```
<owl:Class rdf:about="…">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty
            rdf:resource="…"/>
                <owl:minCardinality
                rdf:datatype="…"> …
…
```

```
<process:AtomicProcess rdf:ID="…">
    <process:hasInput>
        <process:Input rdf:ID="…">
            <process:parameterType
                rdf:resource="…"/>
……
    <process:hasOutput>
……
```

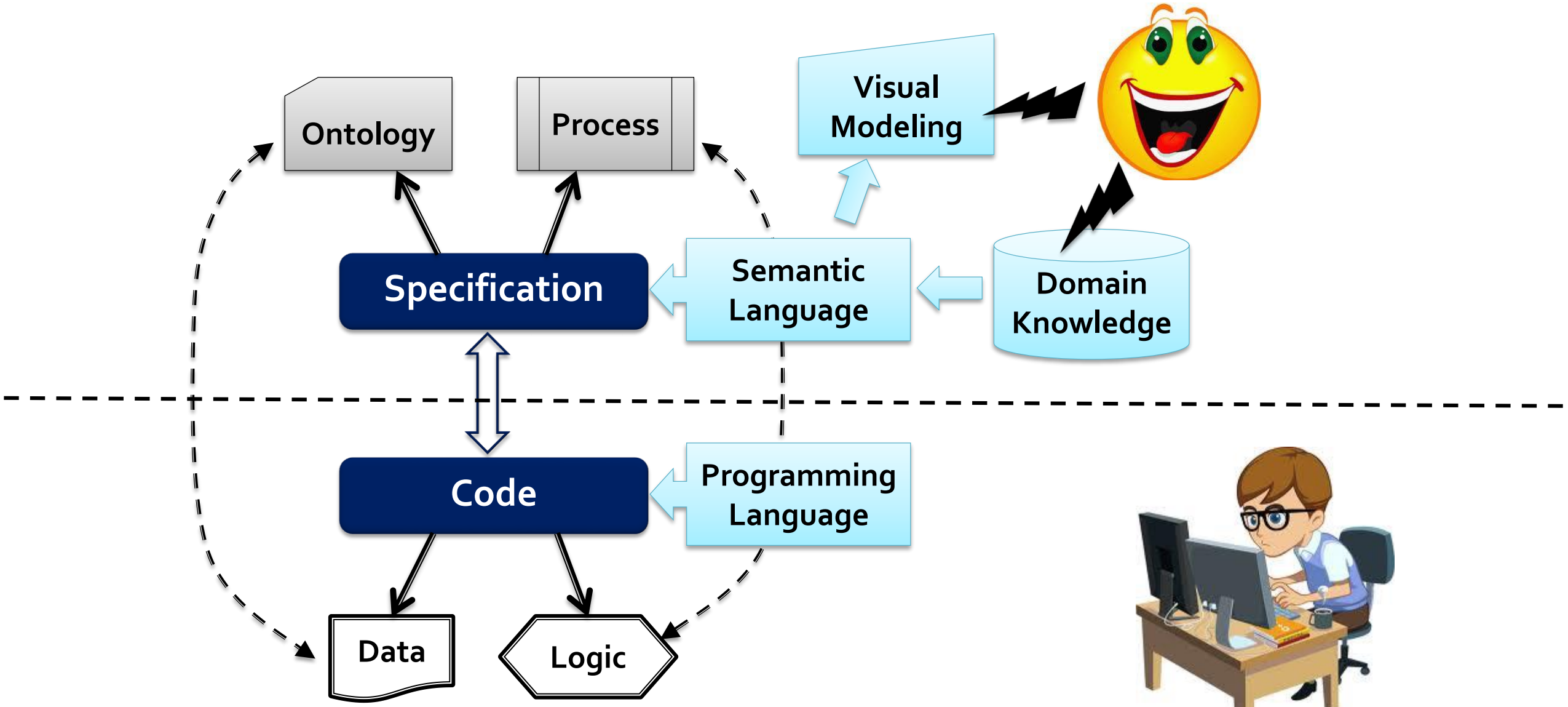
Testing by Contract

- Design by Contract [Meyer 1985]
  - Software components collaborate with each other on the basis of mutual *obligations* and *benefits* which are specified by Interface.

    Pre-condition + Post-condition + Invariant
  - Enhance observability and testability [Briand 2002]
- Testing by contract
  - Contracts represent expected behavior.
  - Contracts are enforced in implementation.
  - Contracts define validation criteria.

From Syntactic to Semantic Understanding

- Ontology-based semantic specifications for SUT behavior understanding
- Domain concepts
  - A common conceptual model for knowledge representation and sharing
  - Data types and their relationships
- Domain functionalities
  - An abstraction of software behavior focusing on the external visible inputs/outputs
  - Pre- and post- conditions


Ontology
Process
Visual Modeling
Specification
Semantic Language
Domain Knowledge
Code
Programming Language
Data
Logic

An Example

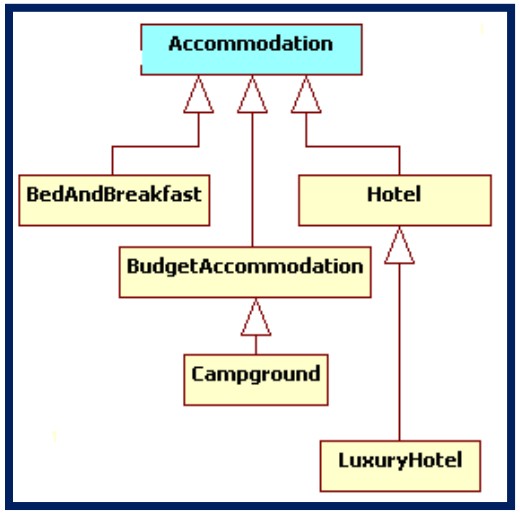
```
<process:process>
  - <process:AtomicProcess rdf:ID="update_FlightInfo_Process">
```

```
<owl:Class rdf:ID="FlightInfo">
  - <rdfs:subClassOf>
    - <owl:Restriction>
        <owl:onProperty rdf:resource="#flightNumber"/>
      - <owl:allValuesFrom>
```

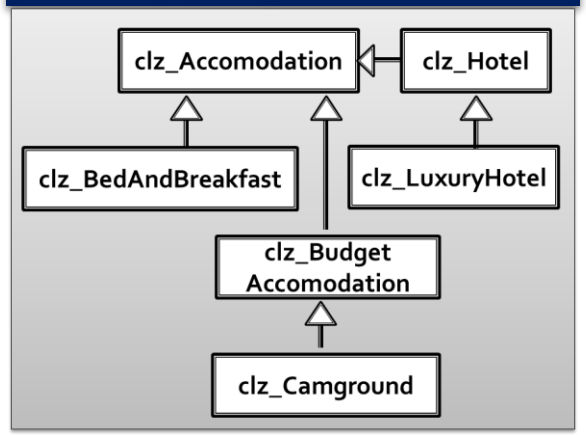
Range

Name	Expression
Rule-1	→ FlightInfo(?x_flight) ∧ FlightInfo(?y_flight) ∧ FlightNumber(?number) ∧ Airline(?x_comp) ∧ Airline(?y_comp) ∧ flightNumber(?x_flight, ?number) ∧ company(?x_flight, ?...
Rule-10	→ FlightInfo(?x) ∧ departureTime(?x, ?dep_time) ∧ arrivalTime(?x, ?arr_time) ∧ after(?dep_time, ?arr_time) → ReturnCode(ARR_BEFORE_DEP_TIME)
Rule-11	→ FlightInfo(?x) ∧ departureTime(?x, ?dep_time) ∧ hasTicketInfo(?x, ?ticket) ∧ departureDateTime(?ticket, ?ticket_date) ∧ swrlb:substring(?ticket_time, ?ticket_date, 11, 5...
Rule-12	→ FlightInfo(?x) ∧ hasCabinInfo(?x, ?cabin) ∧ cabinType(?cabin, BClassCabin) ∧ totalSeats(?cabin, ?total) ∧ hasTicketInfo(?x, ?ticket) ∧ freeSeatsInBCabin(?ticket, ?free...
Rule-13	→ FlightInfo(?x) ∧ hasCabinInfo(?x, ?cabin) ∧ cabinType(?cabin, EClassCabin) ∧ totalSeats(?cabin, ?total) ∧ hasTicketInfo(?x, ?ticket) ∧ freeSeatsInECabin(?ticket, ?free...
Rule-14	→ FlightInfo(?x) ∧ hasCabinInfo(?x, ?cabin) ∧ cabinType(?cabin, FClassCabin) ∧ totalSeats(?cabin, ?total) ∧ hasTicketInfo(?x, ?ticket) ∧ freeSeatsInFCabin(?ticket, ?free...
Rule-2	→ FlightInfo(?x) ∧ departureCity(?x, ?departure) ∧ arrivalCity(?x, ?arrival) ∧ differentFrom(?departure, ?arrival) → ReturnCode(NO_ERROR)
Rule-3	→ FlightInfo(?x) ∧ departureTime(?x, ?dep_time) ∧ arrivalTime(?x, ?arr_time) ∧ after(?arr_time, ?dep_time) → ReturnCode(NO_ERROR)
Rule-4	→ FlightInfo(?x) ∧ departureTime(?x, ?dep_time) ∧ hasTicketInfo(?x, ?ticket) ∧ departureDateTime(?ticket, ?ticket_date) ∧ swrlb:substring(?ticket_time, ?ticket_date, 11, 5...
Rule-5	→ FlightInfo(?x) ∧ hasCabinInfo(?x, ?cabin) ∧ cabinType(?cabin, BClassCabin) ∧ totalSeats(?cabin, ?total) ∧ hasTicketInfo(?x, ?ticket) ∧ freeSeatsInBCabin(?ticket, ?free...
Rule-6	→ FlightInfo(?x) ∧ hasCabinInfo(?x, ?cabin) ∧ cabinType(?cabin, EClassCabin) ∧ totalSeats(?cabin, ?total) ∧ hasTicketInfo(?x, ?ticket) ∧ freeSeatsInECabin(?ticket, ?free...
Rule-7	→ FlightInfo(?x) ∧ hasCabinInfo(?x, ?cabin) ∧ cabinType(?cabin, FClassCabin) ∧ totalSeats(?cabin, ?total) ∧ hasTicketInfo(?x, ?ticket) ∧ freeSeatsInFCabin(?ticket, ?free...
Rule-8	→ FlightInfo(?x_flight) ∧ FlightInfo(?y_flight) ∧ FlightNumber(?number) ∧ Airline(?x_comp) ∧ Airline(?y_comp) ∧ flightNumber(?x_flight, ?number) ∧ company(?x_flight, ?...
Rule-9	→ FlightInfo(?x) ∧ departureCity(?x, ?departure) ∧ arrivalCity(?x, ?arrival) ∧ differentFrom(?departure, ?arrival) → ReturnCode(DEP_AND_ARR_SAME_CITY)

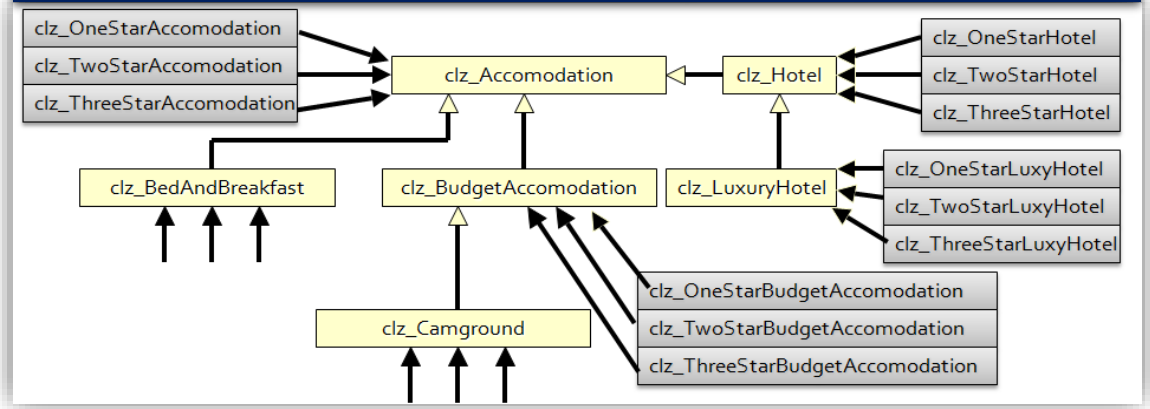
Data Partition Generation



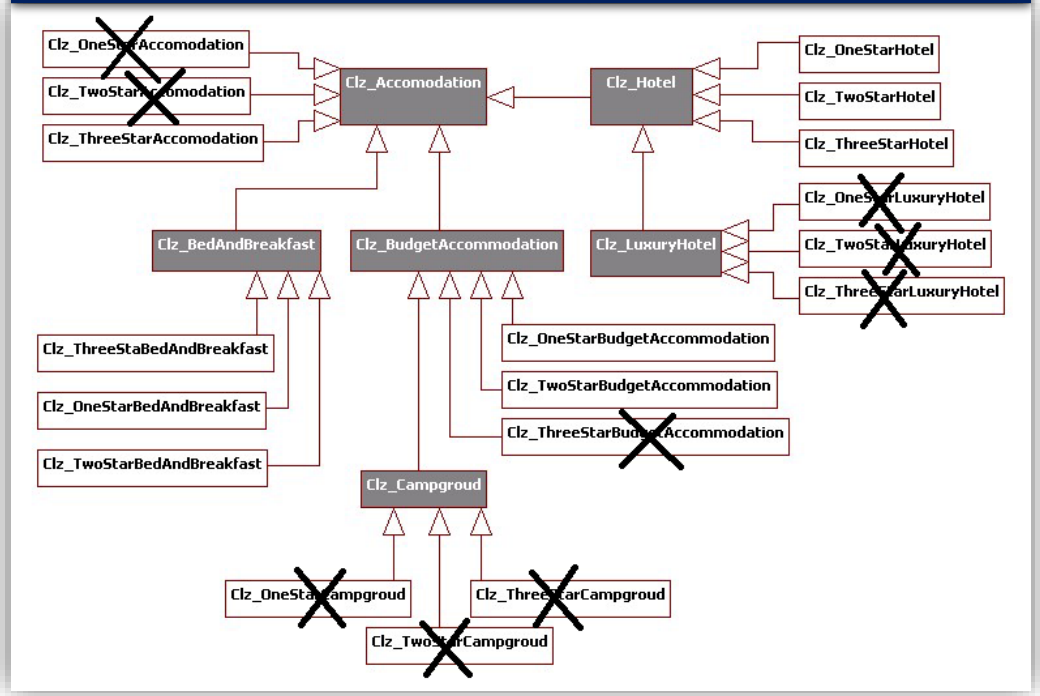
1. Mapping from S to T

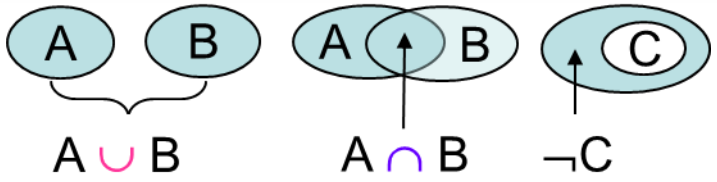


2. Sub-partitions identified by class properties



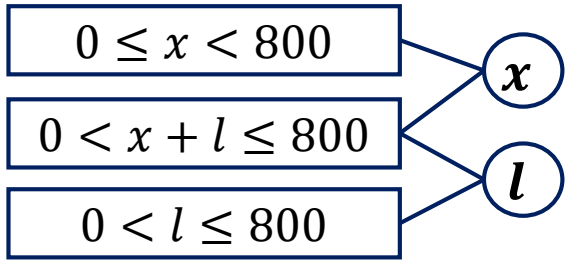
3. Remove redundant class by class relationship and property restriction

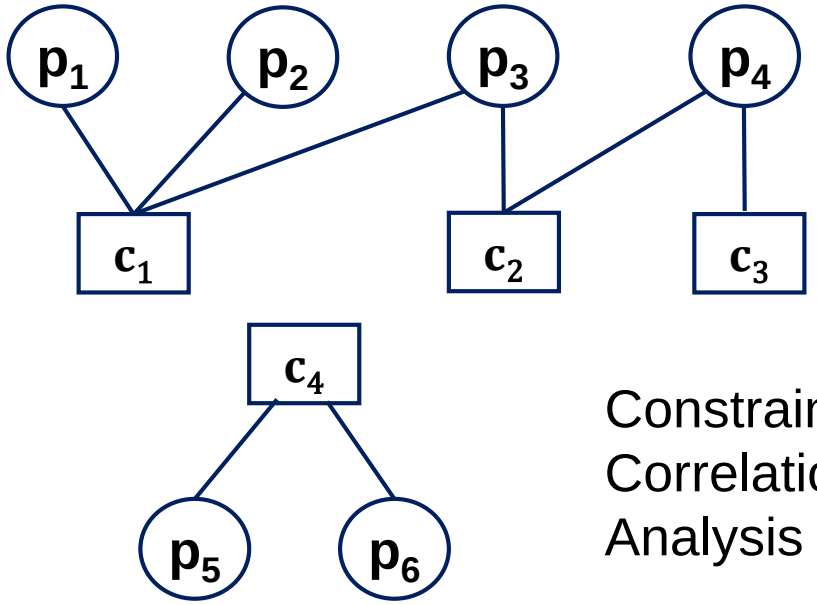




Constraints and Correlations

parameters : x, l
Constraints :
$0 \leq x < 800$
$0 < x + l \leq 800$
$0 < l \leq 800$

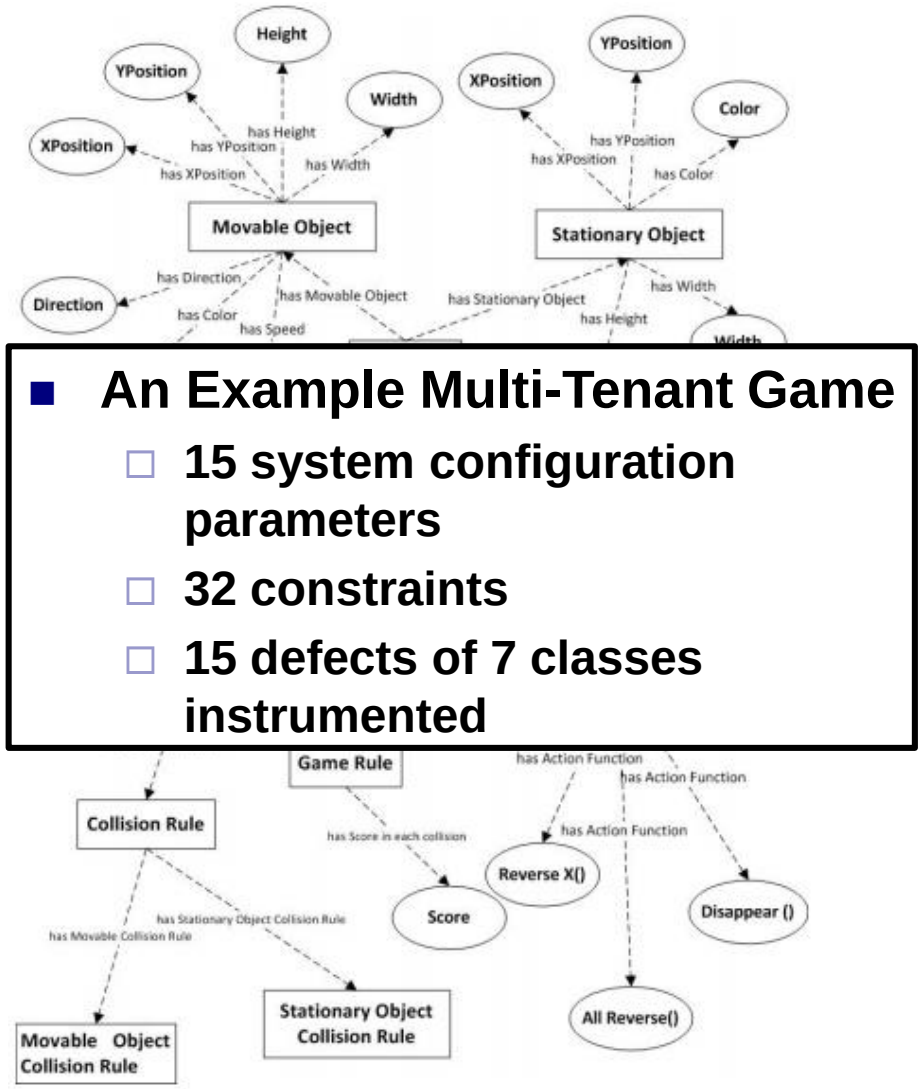




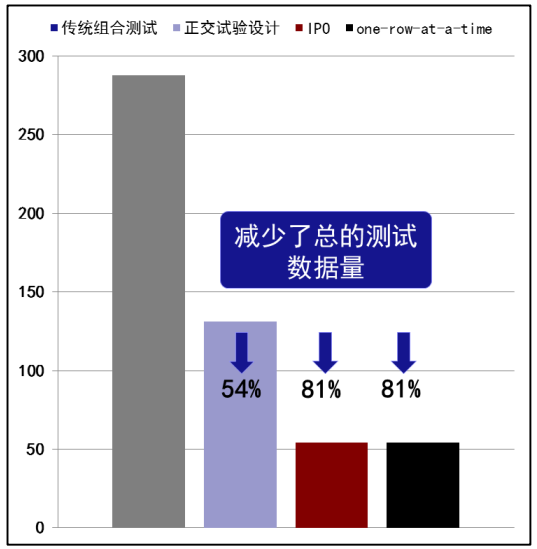
Constraint Correlation Analysis

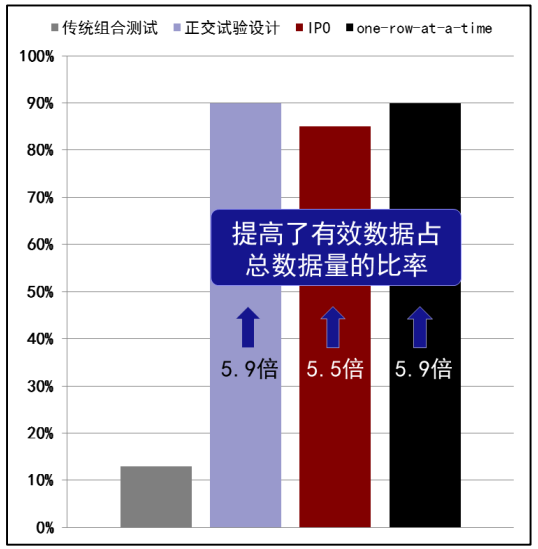
Let G be a graph with no vertices or edges;
FOR (each constraint c_i)
    Generate a vertice vc_i according to constraint c_i
END FOR
FOR (each parameter p_i)
    Generate a vertice vp_i according to parameter p_i
END FOR
FOR (each vertice vc_i)
    IF (constraint c_i contains parameter p_j)
        Then generate an edge e between vc_i and vp_j, e=(vc_i,vp_j)
    END IF
END FOR

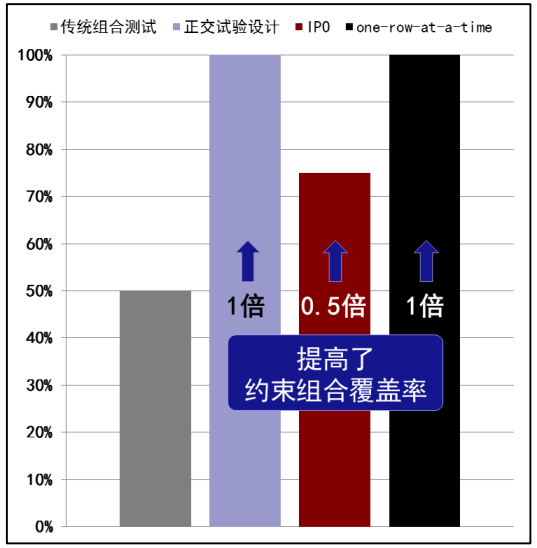
Constraint Combinatorial Testing

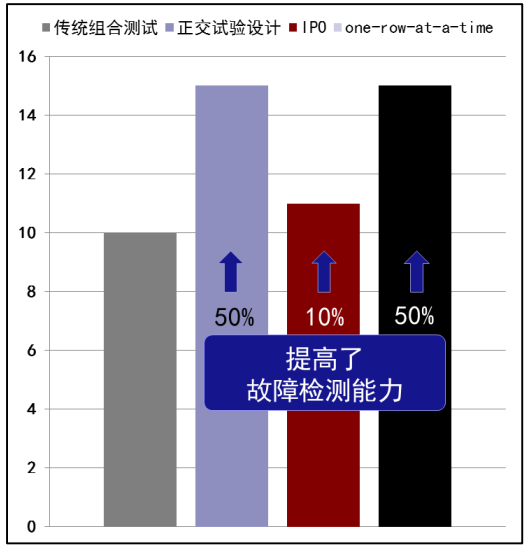


- **An Example Multi-Tenant Game**
  - **15 system configuration parameters**
  - **32 constraints**
  - **15 defects of 7 classes instrumented**









Test Generation by Optimized Search

(Web) API is by nature the executable requirements of software components.

Test Generation

接口名称	输入参数	输出	异常信息	接口说明
激活会员 void activeMember (String memberNo)	memberNo: 会员编号(必填)	无	MemberNotExistsException 会员不存在、待激活的手机或邮箱不存在 MemberBusinessException 会员状态异常(非注册中)	激活注册会员
绑定邮箱 void bindEmail (String memberNo, String email, String loginPassword)	memberNo: 会员编号(必填) loginPassword: 登录密码 (当会员登录密码为空时此参数有效，场景：联合登录会员绑定邮箱时密码为空)	无	MemberExistsException 邮箱已存在异常 MemberBindException 邮箱绑定异常 MemberNotExistsException 会员不存在异常 MemberBusinessException 会员状态异常(正常、业务冻结可绑定邮箱)	如果是修改绑定邮箱，则原有的邮箱失效。原有的邮箱可重新用来注册。

Domain Concepts

Constraints

Machine interpretable

Domain Knowledge

Intelli-gence

Scale

Semantic Modeling

Optimized Search

Optimization for Large-Scale Test Generation

- As problem size and complexity are common challenges to test generation, heuristic search techniques offer promising solutions to cope with the difficulties and optimize test case generation.
- Simulated Annealing (SA) is to simulate the physical annealing process.
  - Objective: the optimized solution at the lowest “temperature”.
  - Search: state switching with decreasing “temperature”
  - A metaheuristic to approximate global optimization in a large search space.

Simulated Annealing with Bayes Classifier

Let $s = s_0$

For $k = 0$ through k_{max} (exclusive):

$T \leftarrow$ temperature (k/k_{max})

Pick a random neighbor, $s_{new} \leftarrow$ neighbor (s)

If $P(E(s), E(s_{new}), T) \geq$ random(0,1), move to the new state

$s \leftarrow s_{new}$

Output: the final sate s

Bayes Classifier

To select the data with the highest potency to detect defects.

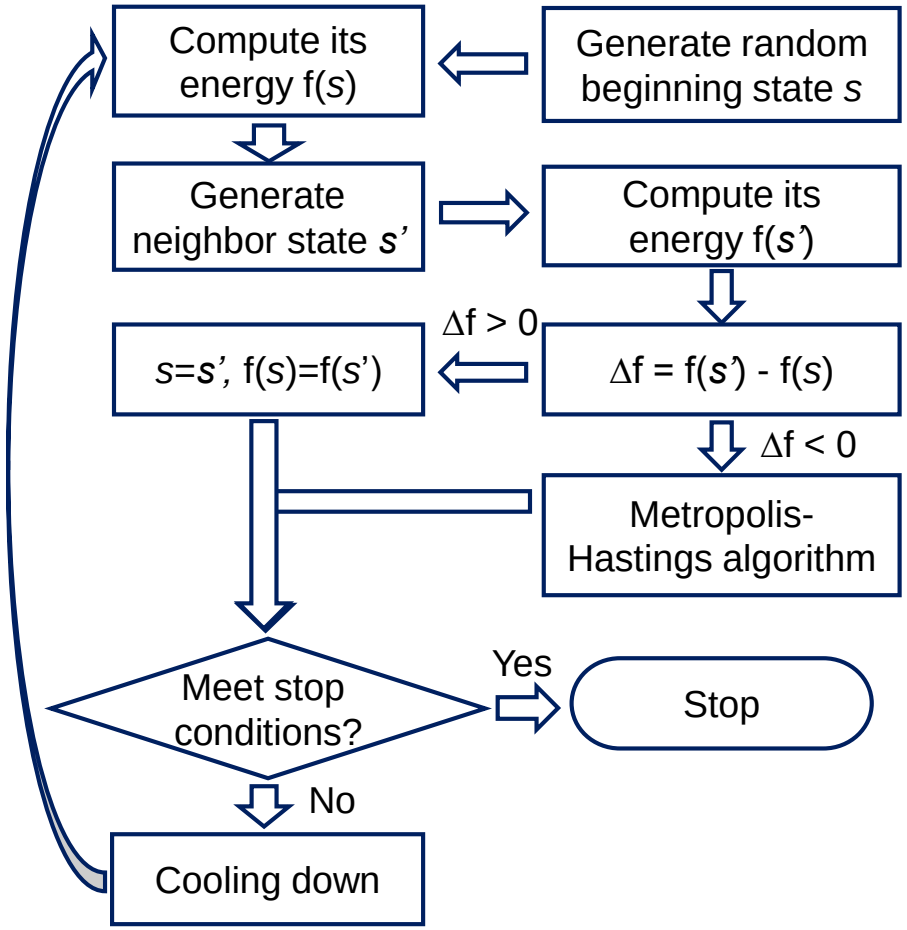
Location Based Service

- LBS can provide information services such as transportation for the given location and has been an enabling technique for a variety of mobile applications.
- Test challenges:
  - Large input space of geographical locations
  - Online evolutions
  - Test oracle

[39.38625,119.3781]	[36.72849,115.2413]	[50.03917，120.0329]
河北省唐山市乐亭县	河北省邯郸市邱县	内蒙古自治区呼伦贝尔市陈巴尔虎旗
无	河北省邯郸市邱县	内蒙古自治区呼伦贝尔市额尔古纳市
无	河北省邯郸市邱县梁二庄镇	内蒙古自治区呼伦贝尔市陈巴尔虎旗
河北省唐山市乐亭县	河北省邯郸市邱县	内蒙古自治区陈巴尔虎旗

AMAP

Search Test Data by SA



Search Test Data with High Potency to Detect Defects

Defect Clustering Assumption

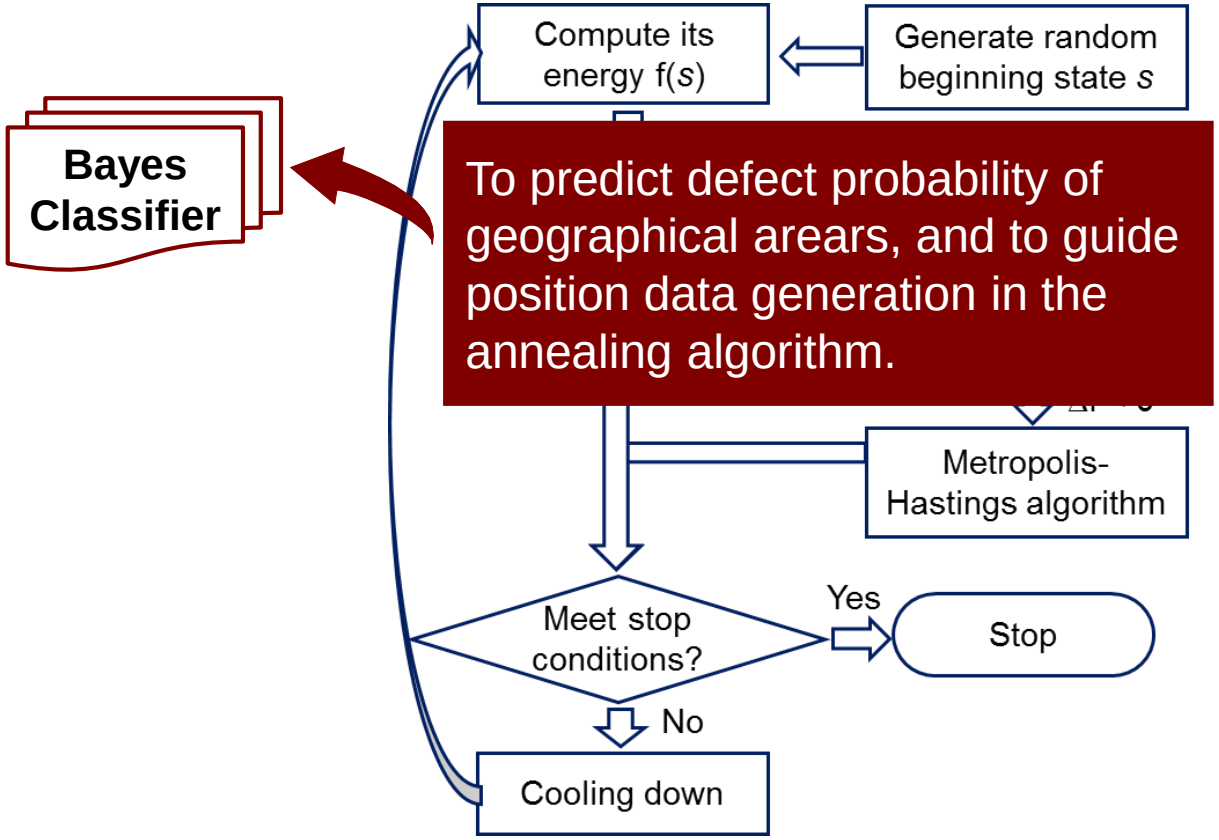
Defects intend to cluster in certain areas. Hence, areas with detected defects deserve more test cases in the follow-up testing.

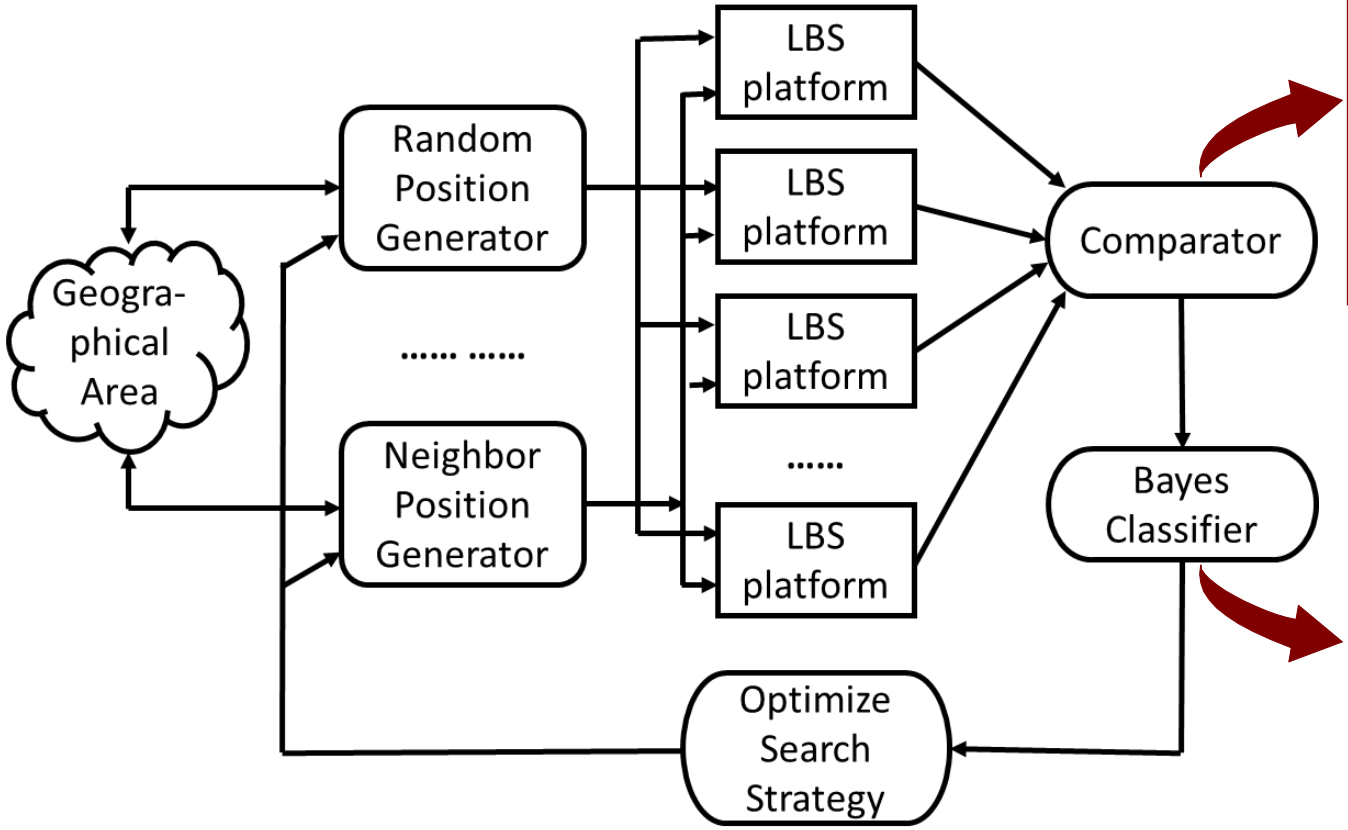
Characteristic Function

To estimate potential errors for testing a position in the geographical area.

Naïve Bayes Classifier

The error-proneness of an area *D* is evaluated by the posterior probability for a potential error detected in the area.





$$Hit(s_i, L) = \begin{cases} 1, & \text{if } \exists l_{i1}, l_{i2} \in L \text{ such that} \\ & R(L_{i1}, s_i) \neq R(L_{i2}, s_i) \\ 0, & otherwise \end{cases}$$

$$Hit(S^D, L) = \sum_{\forall s_i \in S^D} Hit(s_i, L)$$

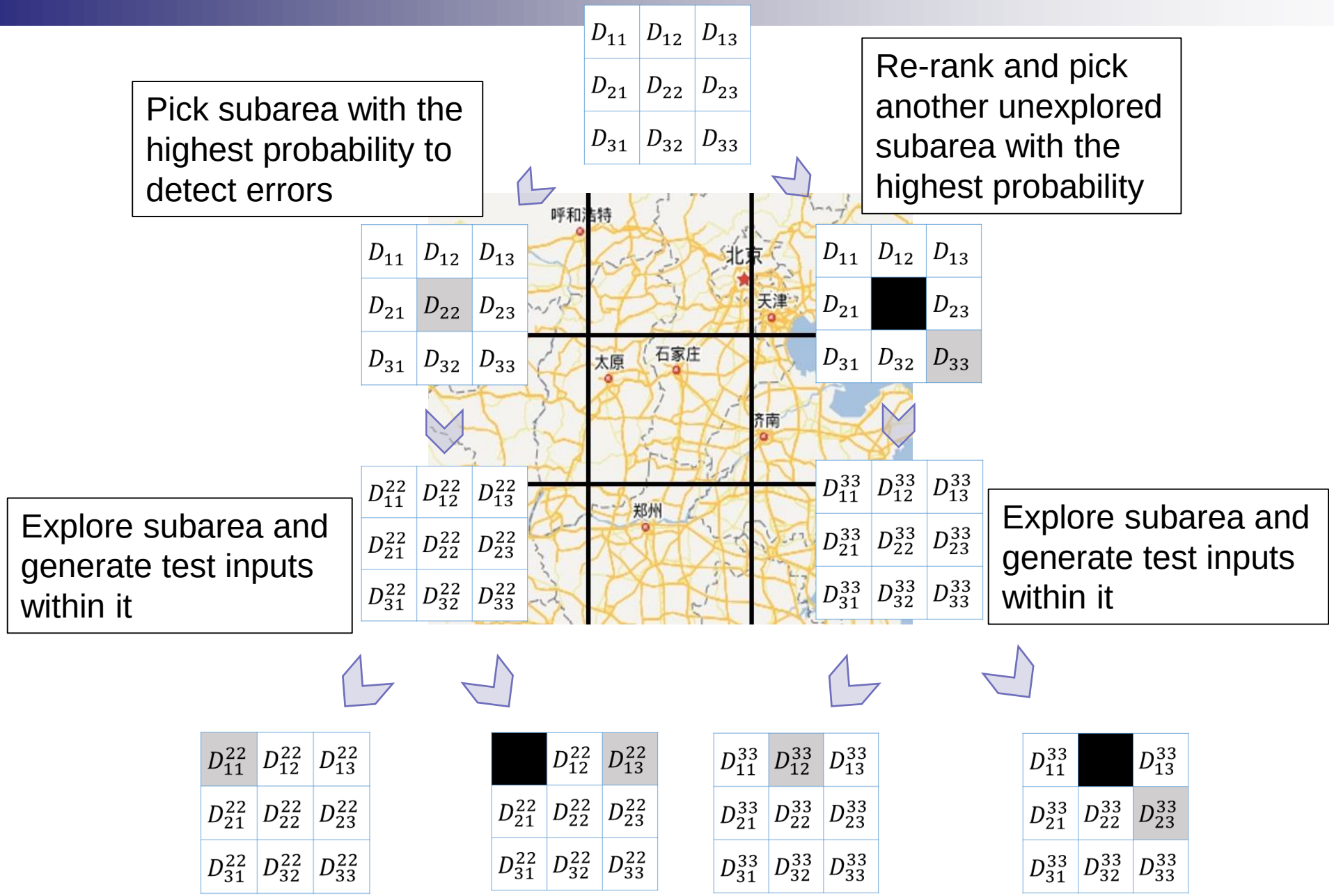
$$P(D_{i,j} | Hit(s_k, L) = 1) = \frac{Hit(S^{D_{i,j}}, L)}{Hit(S^D, L)}$$

$$P(Hit(s_k, L) = 1) = \frac{Hit(S^D, L)}{|S^D|}$$

$$P(D_{i,j}) = \frac{|S^{D_{i,j}}|}{|S^D|}$$

$$P(Hit(s_k, L) = 1 | D_{i,j}) = \frac{P(D_{i,j} | Hit(s_k, L) = 1 \times P(Hit(s_k, L) = 1))}{|P(D_{i,j})|}$$

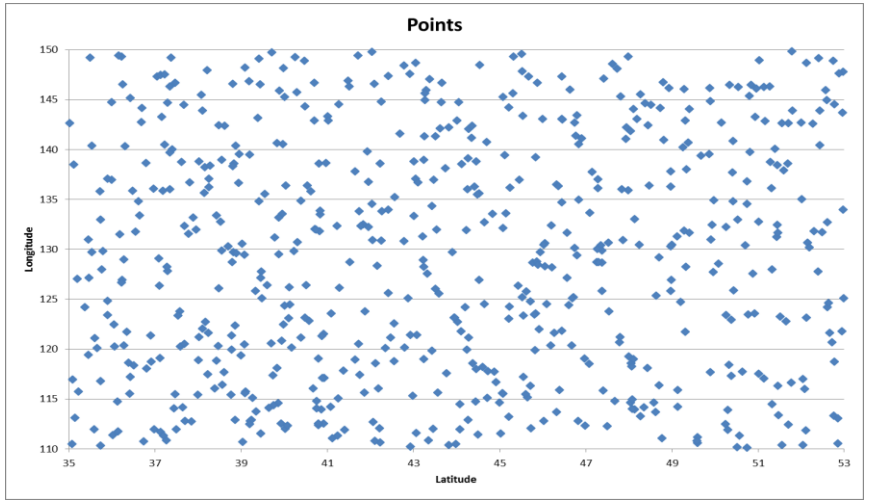
Approach Overview

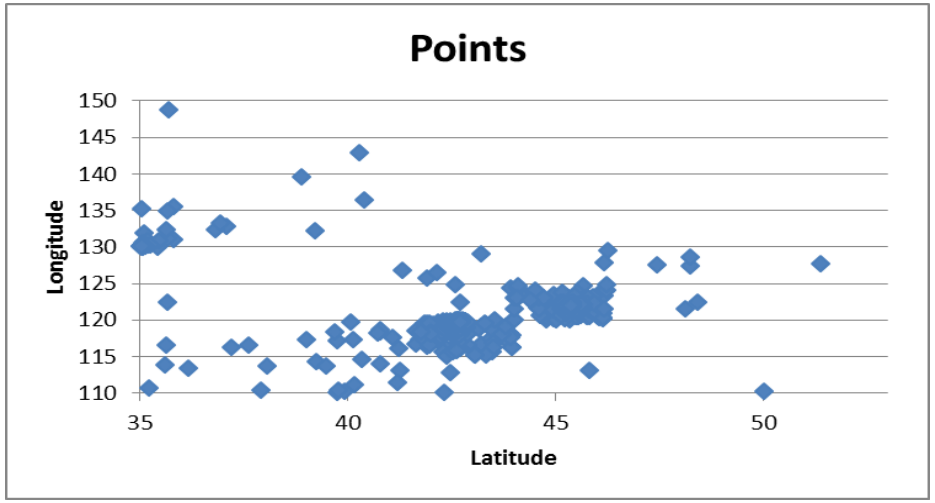

Pick subarea with the highest probability to detect errors
Re-rank and pick another unexplored subarea with the highest probability
Explore subarea and generate test inputs within it
Explore subarea and generate test inputs within it
呼和浩特
北京
天津
太原
石家庄
济南
郑州

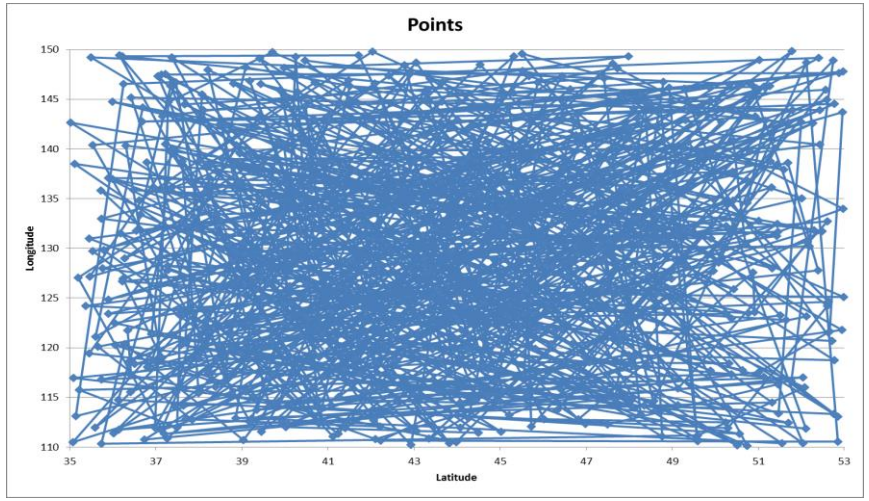
Experiments

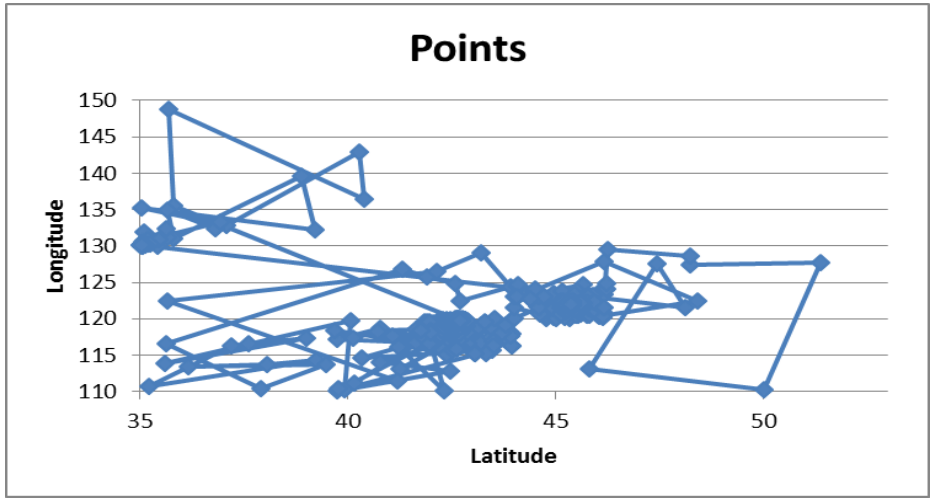
- Target
  - 4 different LBS platforms: Amap Android, Tencent Android, Baidu Android, and Baidu Web
- SA control
  - Initial temperature=250, and temperature threshold=0.05
  - Cooling down factor=0.98
- Test case size control
  - Test case size 3600
  - $Threshold^{D}$ =100, $Threshold^{D\_state}$ =10
- Area under test:
  - D.rangeLat=[35, 53], D.rangeLng=[110, 150]
  - Range of latitude and longitude should be larger than 0.05

Test Data Generation



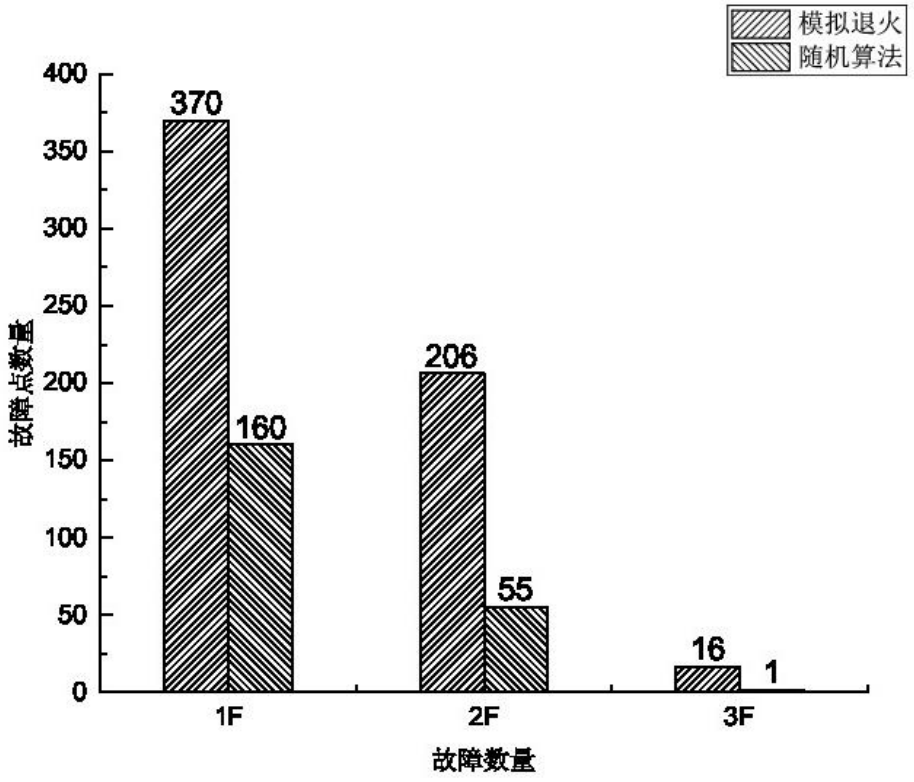


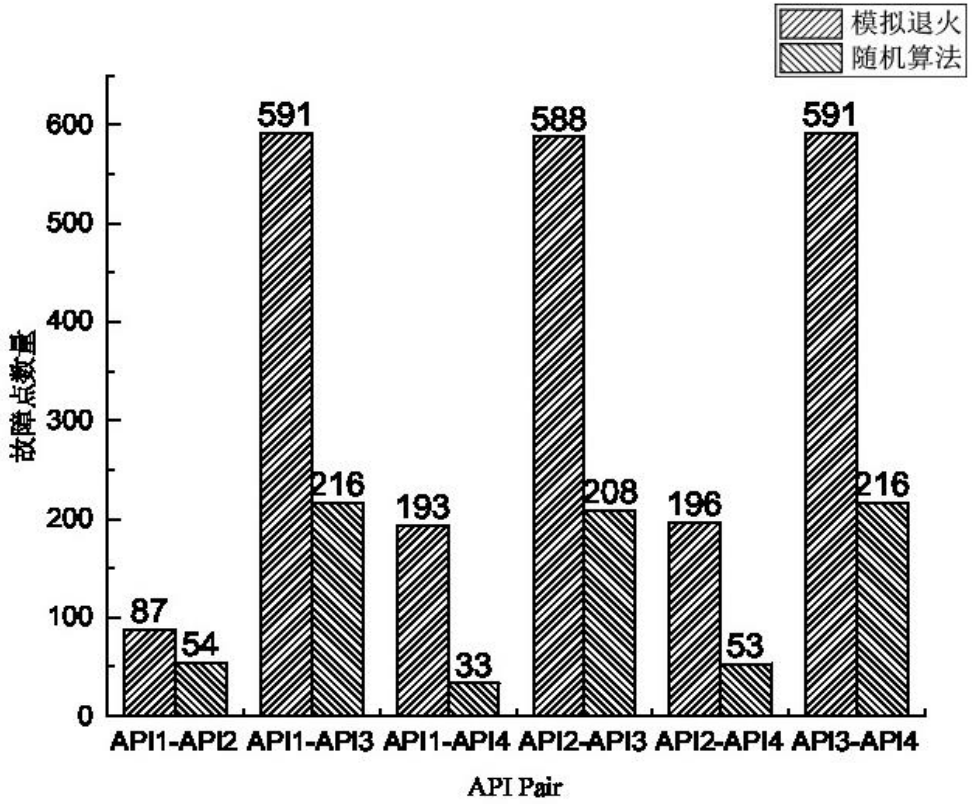




Position data generated by SA are more clustered.

Defect Detection





Comparison of potential errors detected by improved SA and Random.

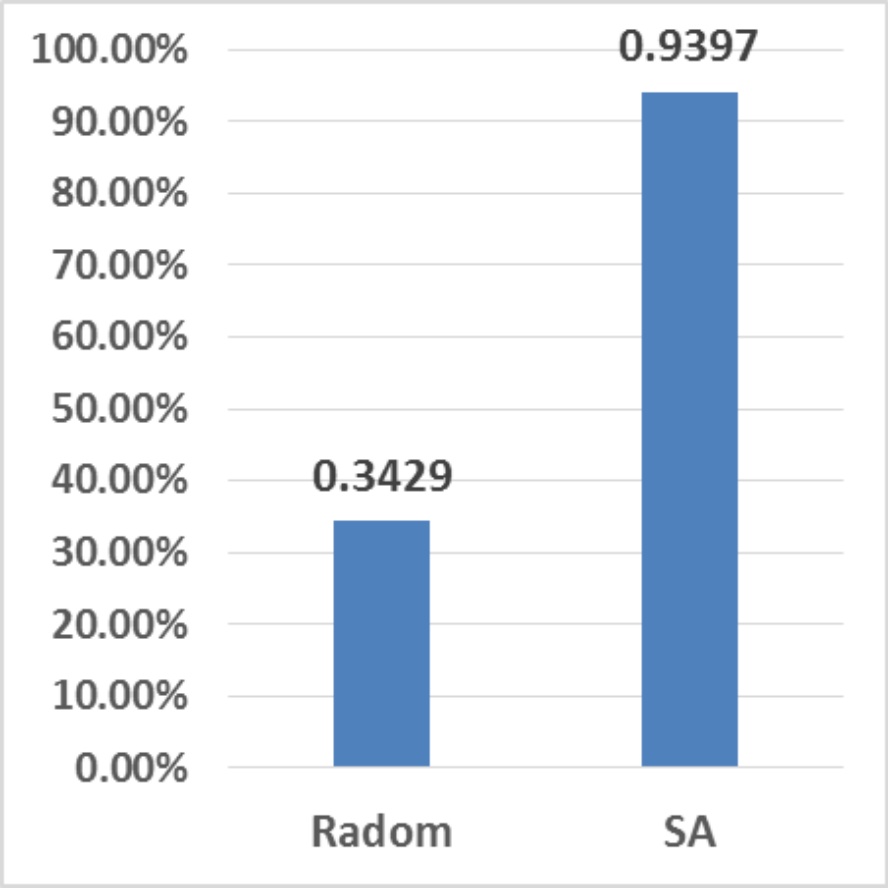
Efficiency

$$TE = \frac{|S_{diff}|}{|S|} \times 100\%$$

Where

- $S = s_i$ is the set of all generated test cases, and $|S|$ is the number of generated test cases.

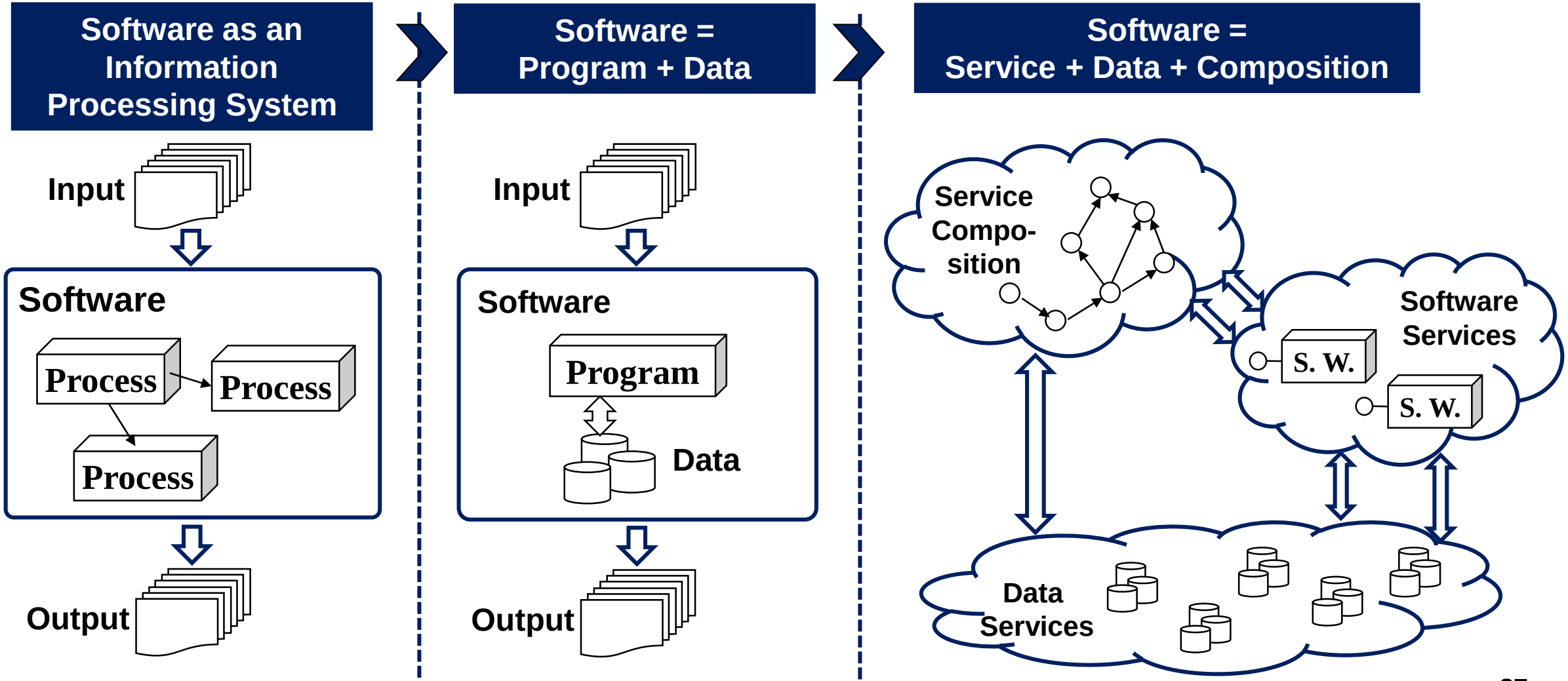
- $S_{diff} = \{s_j\}$ is the set of test cases which identified API differences, that is, $\forall s_j \in S_{diff}$, $s_j \in S$ and $Hit(s_j, L)=1$. $|S_{diff}|$ is the size of S_{diff}

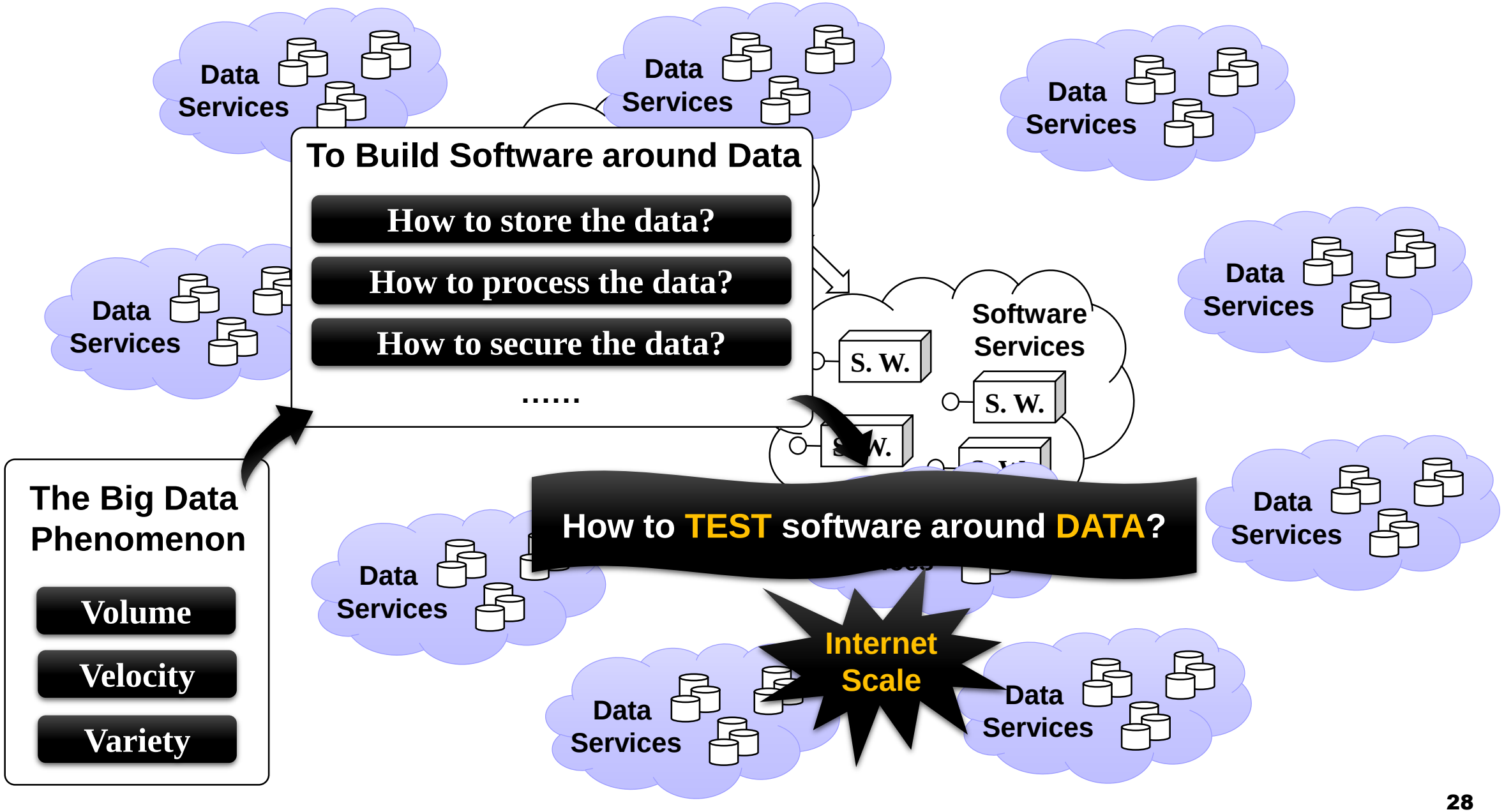


Summary

- Considerable inconsistencies are detected on open LBS platforms, which are potential errors or quality problems such as accuracy, completeness, and up-to-dateness.
- Geographic data generation can be well-formulated as an optimized search problem. The proposed SA with defect prediction mechanism can significantly enhance the effectiveness of test data generation.

Paradigm Shift




Data Services
Data Services
Data Services
Data Services
Data Services
Data Services
Data Services
Data Services
Data Services
To Build Software around Data
How to store the data?
How to process the data?
How to secure the data?
.....
Software Services
S. W.
S. W.
The Big Data Phenomenon
Volume
Velocity
Variety
How to TEST software around DATA?
Internet Scale

Thank you!

Xiaoying Bai

Department of Computer Science and Technology

Tsinghua University

Email: baixy@tsinghua.edu.cn

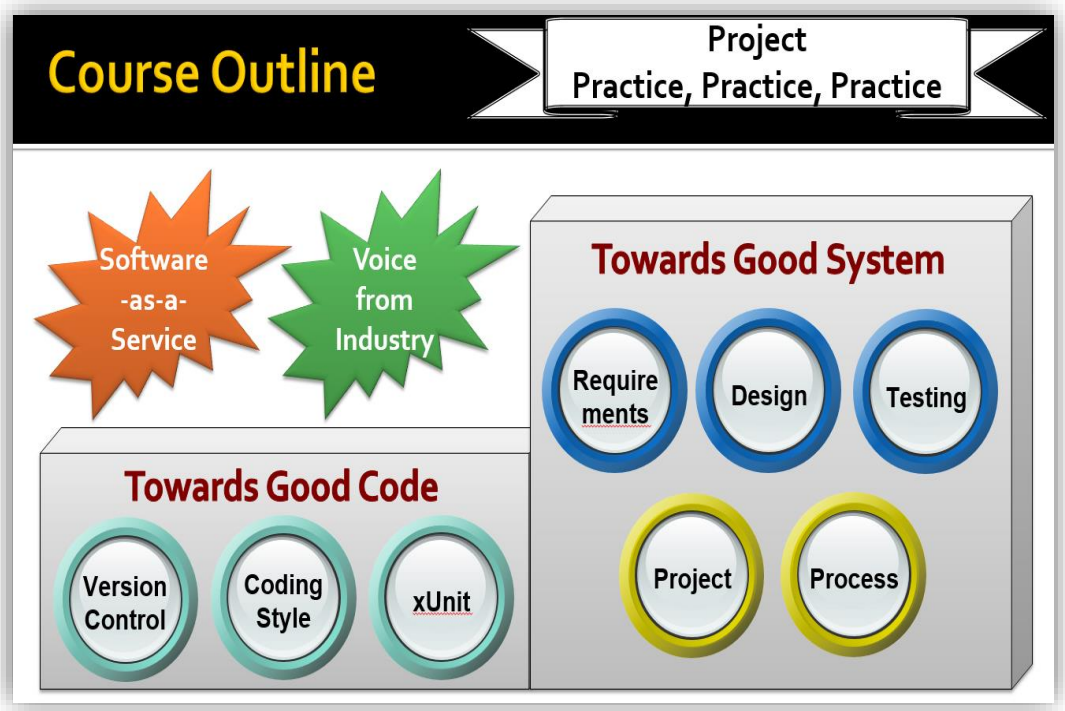
IPIT Collaboration – Software Engineering Education and Research

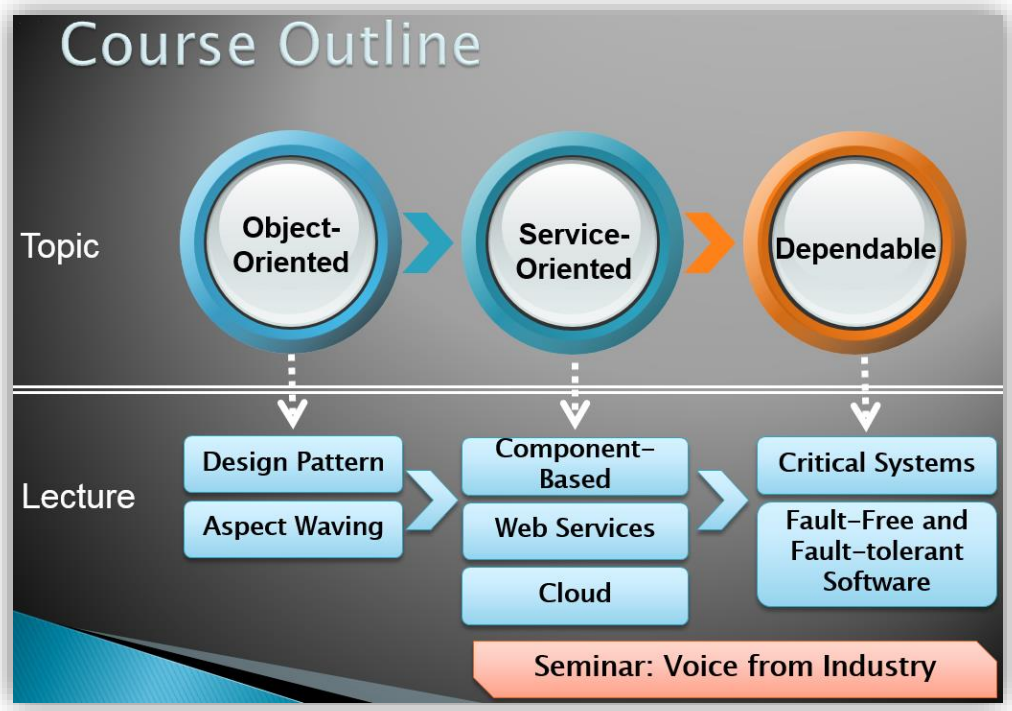
Xiaoying Bai
Department of Computer Science and Technology
Tsinghua University
June, 2018

About Me

- Experience
  - 2006.1 – present, Associate professor, Dept. of Com. Sci & Tech., THU , China
  - 2002.1 – 2005.12, Assistant professor, Dept. of Com. Sci & Tech., THU, China
  - 2002.4 – 2008.12, Technical expert, BOCOG
- Education
  - 2000.1 – 2001.12 , Ph.D, Dept. of Comp. Sci. & Tech., ASU, US,
    Advisor: Dr. W.T. Tsai
  - 1998.9 – 1999.12, Ph.D candidate, Dept. of Comp. Sci. & Tech., UMN, US,
    Advisor: Dr. W.T. Tsai
  - 1995.9 – 1998.4, M.S., Dept. of Comp. Sci. & Tech., BUAA, China
    Advisor: Dr. Wei Li
  - 1991.9 – 1995.7, B.S., Dept. of Comp. Sci. & Tech., NPU, China

About Me: Teaching

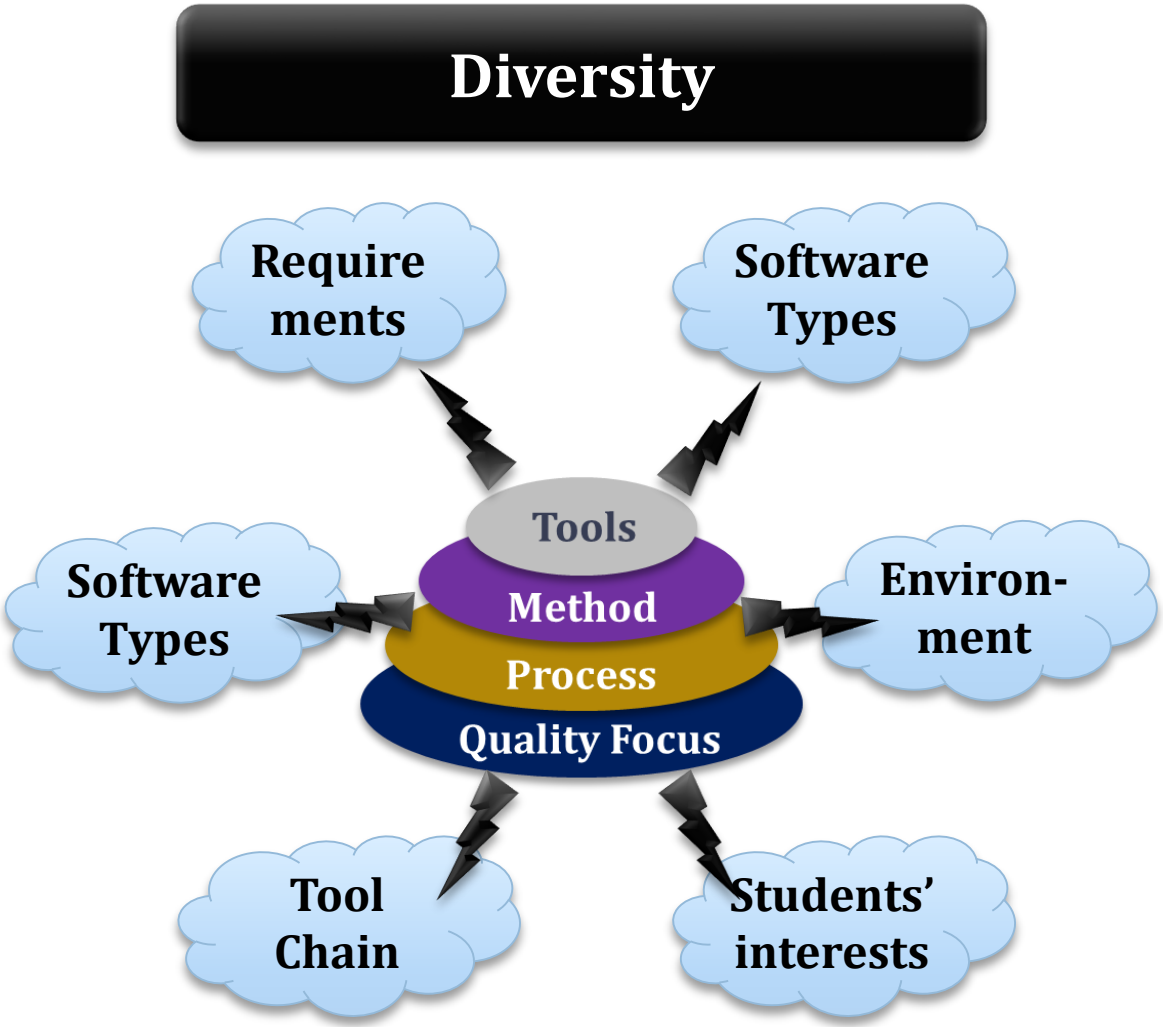


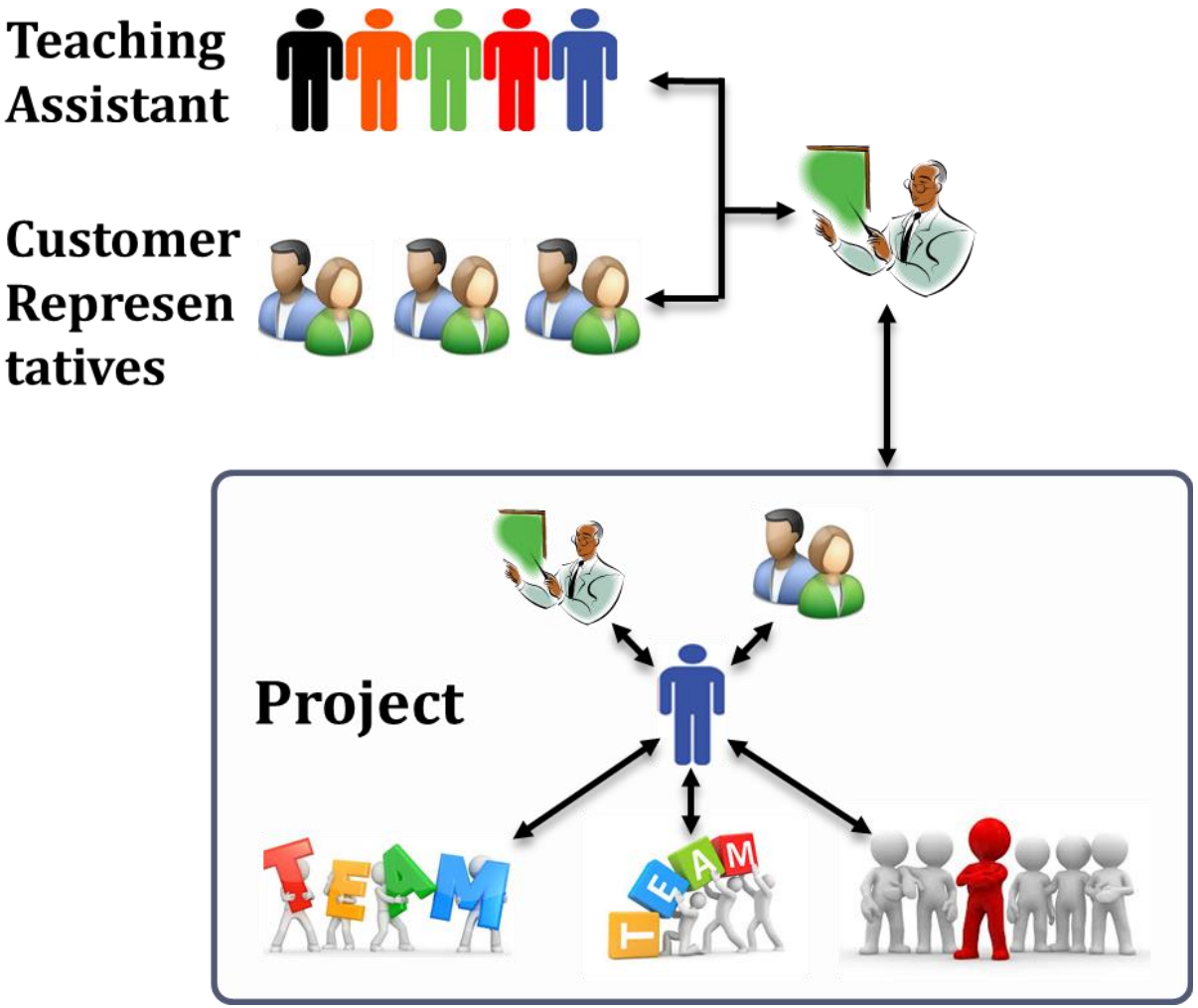


Undergraduate:
Introduction to Software Engineering

Graduate:
Advanced Software Design Techniques

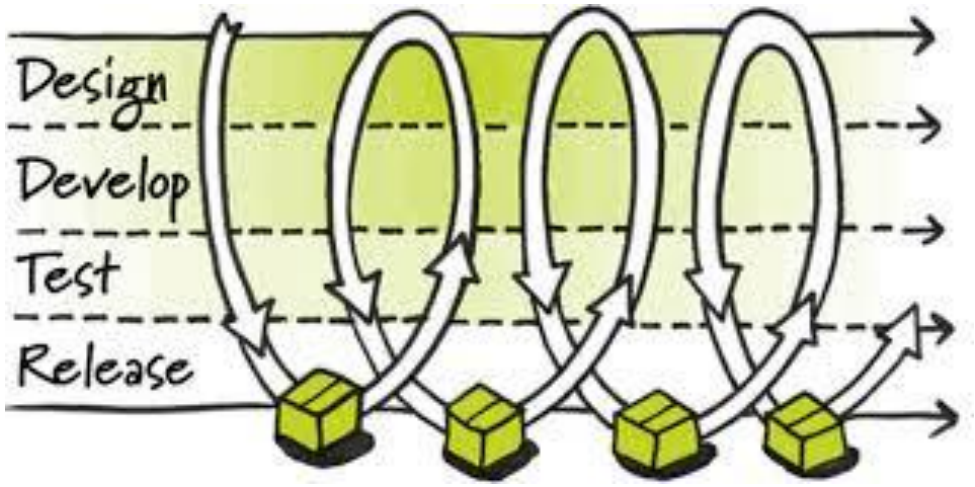
To Renovate SE Course Projects



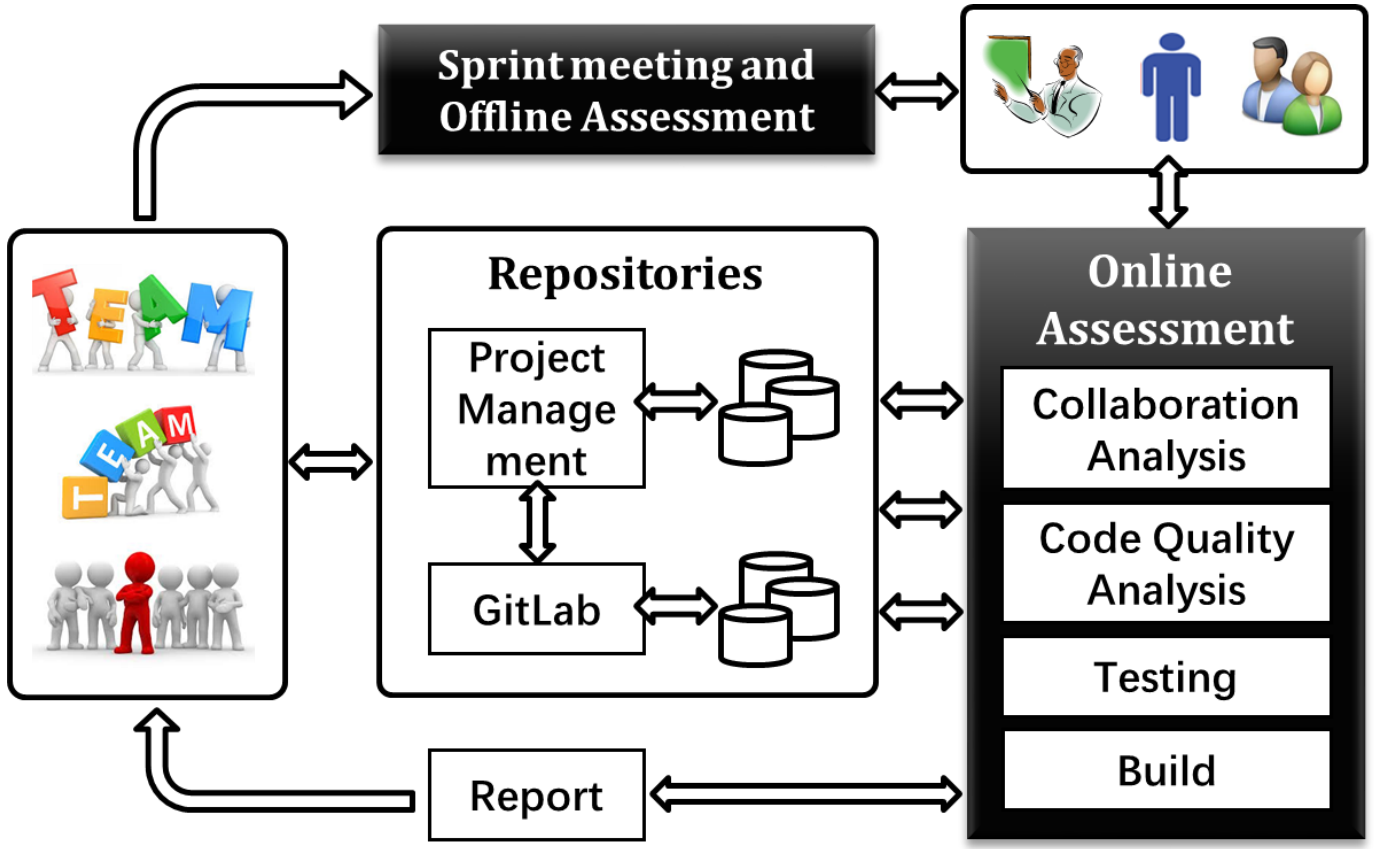


To Renovate SE Course Projects

Agile



- Small, incremental delivery of working software
- Continuous integration, testing, and quality control



Thank you!

Xiaoying Bai

Department of Computer Science and Technology

Tsinghua University

Email: baixy@tsinghua.edu.cn