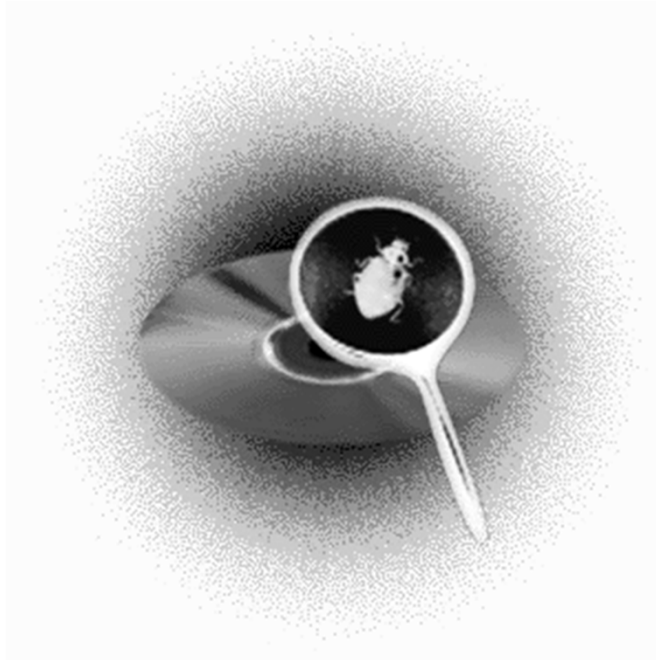


Software Test Architecture Design focusing on Test Viewpoints



SOFTEC 2012

2012/7/12 (Thu)

Nishi, Yasuharu

The University of Electro-Communications, Japan

© NISHI, Yasuharu

Profile

Assistant professor:

the University of Electro-Communications, Japan

(also providing consultancy service to industry on testing and TQM)

President:

Association of Software Test Engineering, Japan (ASTER)

President:

Japan Software Testing Qualifications Board (JSTQB)

National delegate:

ISO/IEC JTC1/SC7/WG26 Software testing

Founder:

Japan Symposium on Software Testing (JaSST)

Founder:

Testing Engineers' Forum (Japanese community on software testing)

Vice chair:

SQiP/Software Quality Committee of JUSE (promoting organization of TQM)

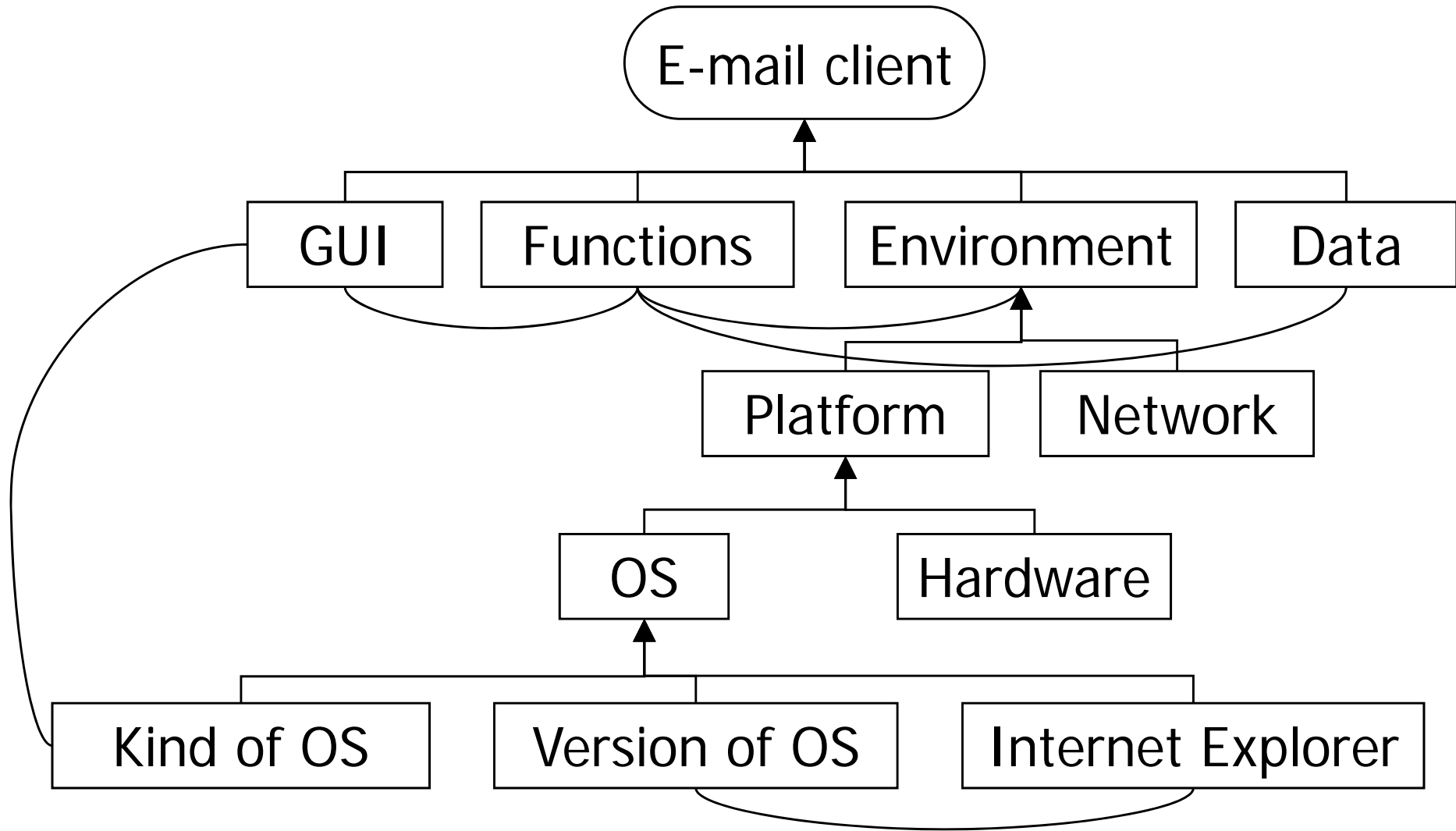
(SQiP has published the book of "SQuBOK: Software Quality Body of Knowledge" and is operating engineer certification on software quality)



Can you grasp a big picture of test design?

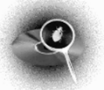
- Which expression your organization use?
 - Unnecessary, because of the same as specs
 - » Test engineers must design test cases BEYOND specs because specs are always not enough, ambiguous and faulty
 - List of Software quality characteristics or non functional requirements
 - (Master) test plan
 - Abstract description of test
 - List of test types
 - Phases or levels of testing (incl. regression test)
 - List of test models
 - List of test techniques
 - List of Quality risks
 - etc

Example diagram of test architecture



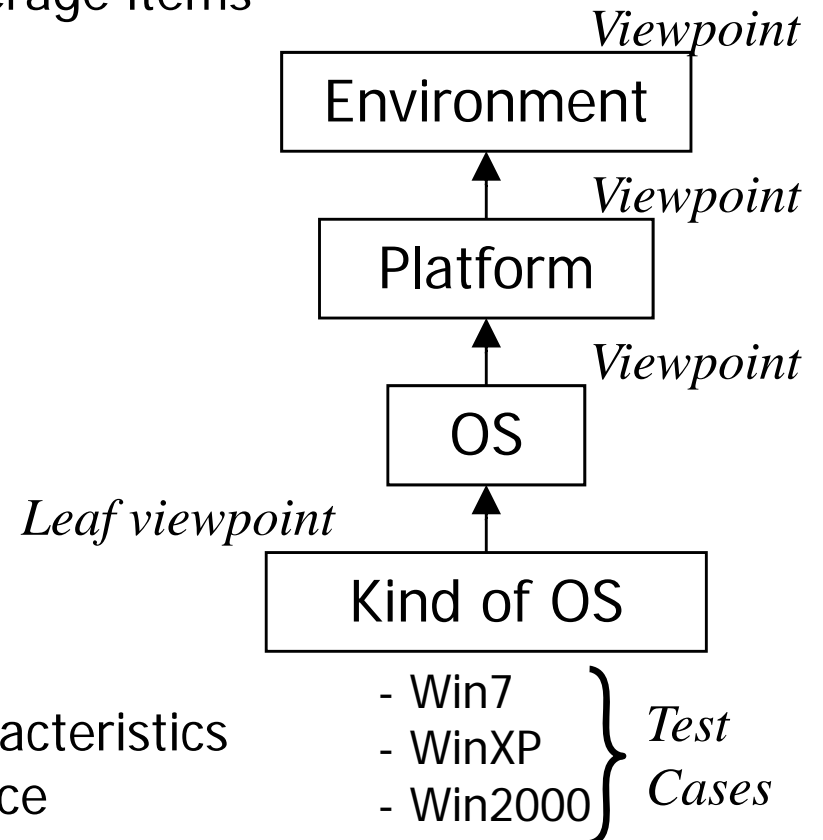
Test architecture focusing on test viewpoints

- Test architecture is a big picture of test design
 - Test engineers have to grasp a big picture of test design because test cases increase over 100,000 cases and get much complicated
 - Test techniques and coverages cannot prevent large lacks of test cases though they can prevent small lacks of test cases
 - Quality of test design depend more on total balance than priority of each test case
- Test architecture is just architecture of test design
 - In software testing domain, people confuses big pictures of test design and big pictures of test process or test management
 - » In software development, software architecture is not described in project plan though test architecture is described in test plan
 - What kinds of test you design should be prior to order of test cases
- Test architecture consists of “test viewpoints” and relationships of them
 - NGT (Notation for Generic Testing) is a made-in-Japan hierarchical notation of test architecture
 - Some experts use mind-maps



Test viewpoint as abstract test cases

- Test cases has test parameters and values
 - ex) parameter: Kind of OS, values: Win7, WinXP, Win2000
 - Test parameters are also called as test conditions and test values are also called as test coverage items
 - Test cases consists of test values
- Viewpoints are abstract test cases
 - Leaf viewpoints means test parameters
 - Viewpoints don't express any test values or test cases
 - Viewpoints can have hierarchically abstract viewpoint
 - » Like classification trees or class diagrams
 - Viewpoints can abstract test conditions, test items and quality characteristics such as load, configuration and performance



Various test viewpoints

- Test viewpoint is a point where test engineers focus an attention for grasping a big picture of test design
 - Test viewpoint is abstraction and source of test cases
- Types of test viewpoints depend on organizations and/or test engineers

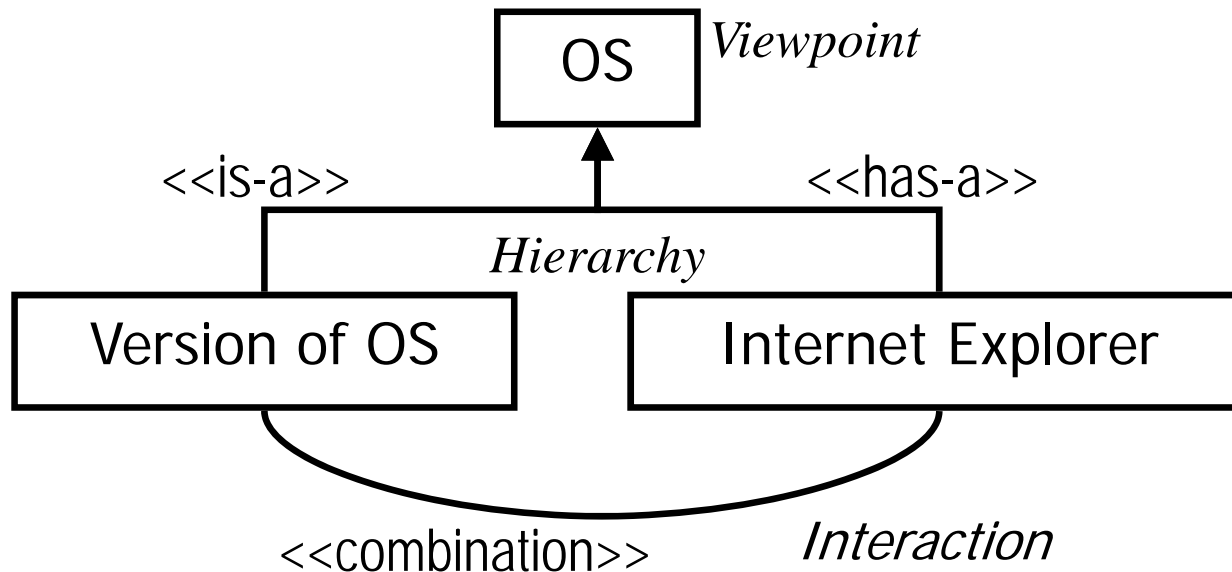
OS

- What should be exhausted:
 - » Specs, functions, data etc.
 - » Test conditions
- Characteristics which should be achieved
 - » Quality characteristics, non functional requirements etc.
- Parts of test items
 - » Funcs, Subsystems, modules etc.
- Bugs
 - » Errors and failures, bug patterns, weak points of test items etc.
- Customer usage
 - » Business, lifestyle etc.
- Other parts of systems than software
 - » Hardware units, hardware failures etc.
- Test types
 - » Load test, configuration test etc.
- Test levels
 - » Component test, system test etc.
- Lists and/or diagrams developed until software testing
 - » Use cases, State transition diagrams etc.

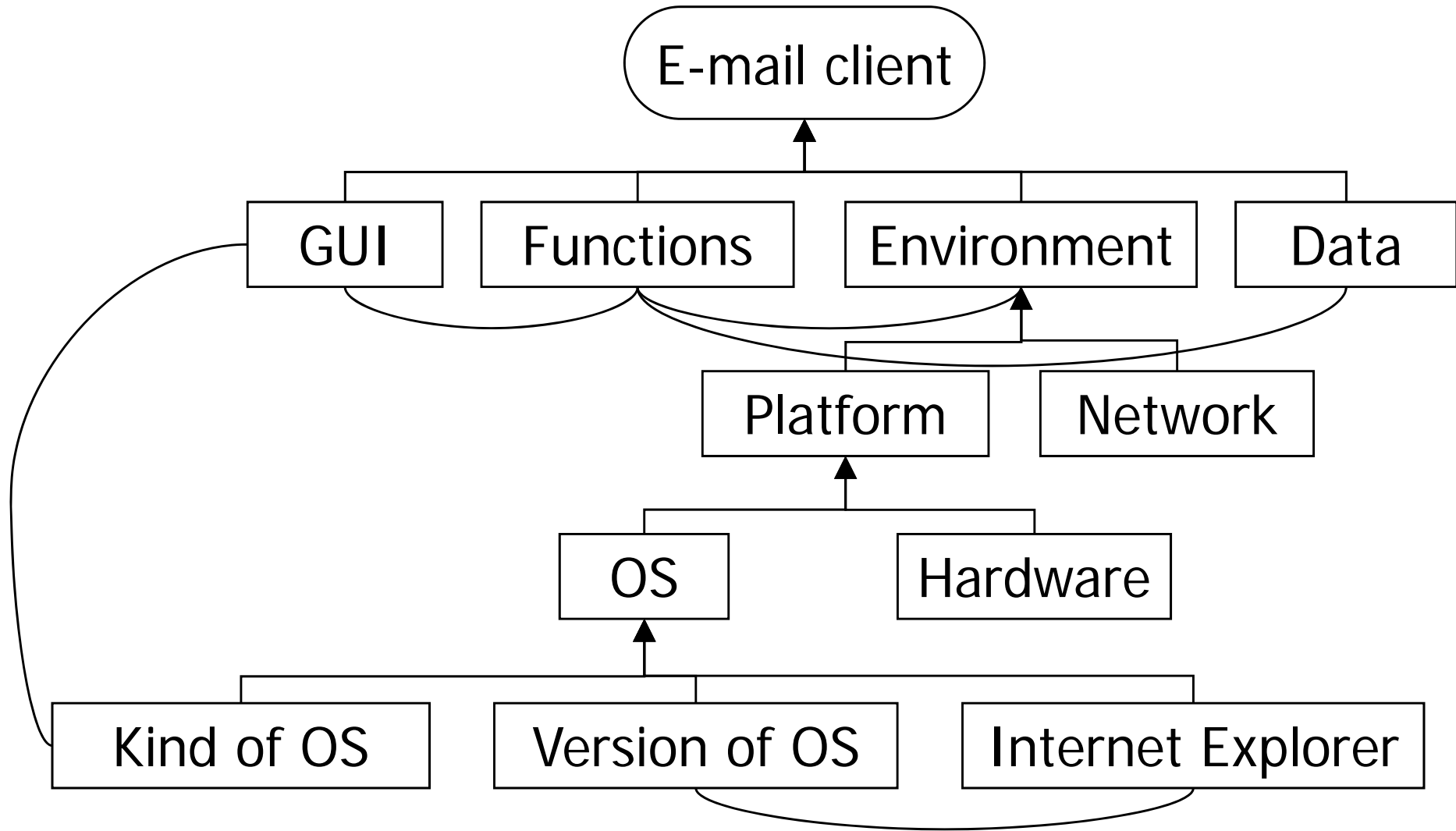


Relationships of test viewpoints

- Test viewpoints have two fundamental relationships
 - Hierarchy relationships
 - » Detail a viewpoint step by step to reach test coverage item with a straight line
 - » Have several types such as is-a, has-a, cause-effect, object-attribute
 - Interaction relationships
 - » Connect test viewpoints to test combination of viewpoints with a curved line
 - » Have several types such as combination (needs combinatorial testing) etc.
- Types of relationships can be expressed as “<<stereotype>>”

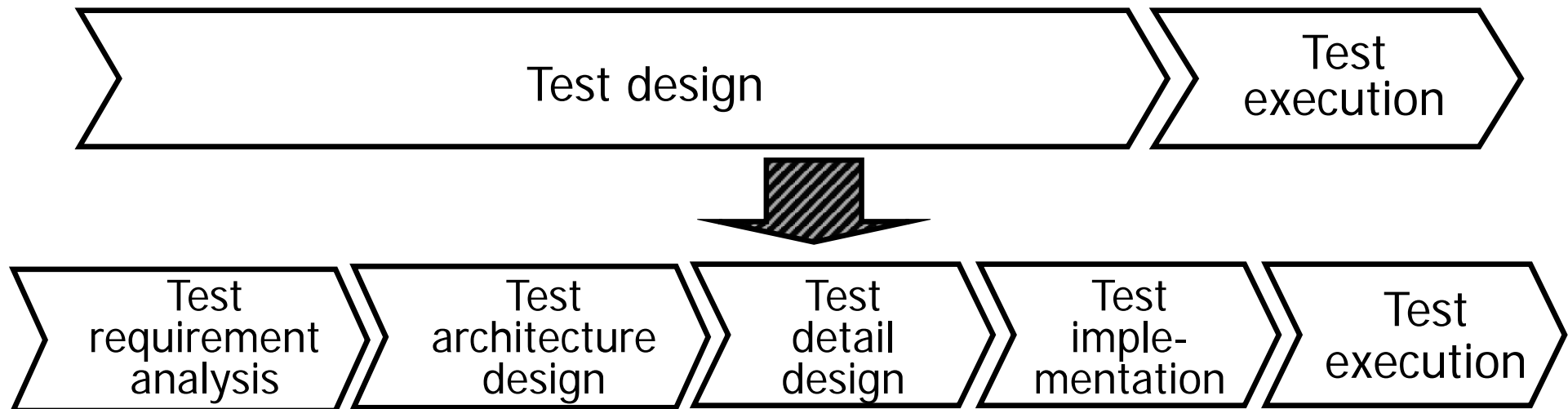


Example diagram of test architecture



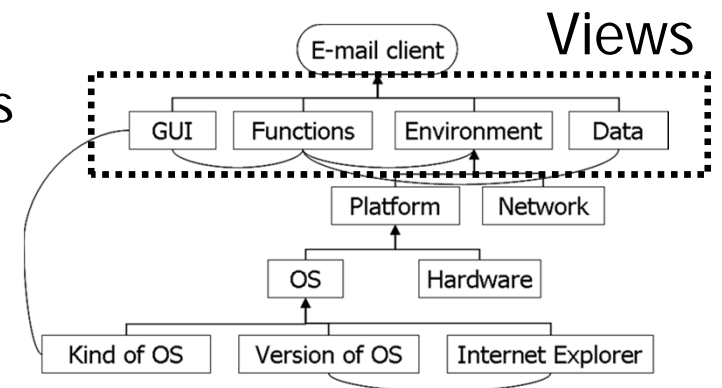
Test process focusing on test viewpoints

- VSTeP (Viewpoint-based Software Testing Process) is a test process focusing on test viewpoints
 - Test requirement analysis
 - Test architecture design
 - Test detail design
 - Test implementation
 - Test execution
 - Test log recording
 - Test result analysis
 - Test improvement
 - Product quality reporting



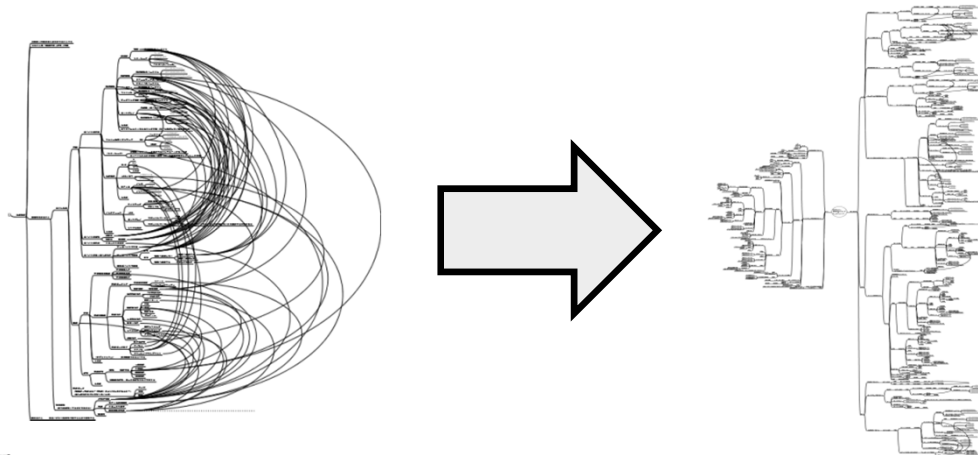
Test requirement analysis

- To extract, organize and understand test requirement
 - Requirements from customers to achieve
 - » Functional requirement, non-functional requirement, business goals etc.
 - Constraints to achieve requirement from customers
 - » Requirement of test project management such as efforts, costs etc.
 - » Test tools and/or methods directly requested by customer especially
 - Information of current quality of the test item
 - » Ex) bugs which were detected in prior reviews
- To create a test requirement model
 - Extract test viewpoints from test requirements
 - Detail test viewpoints and connect parent viewpoint and child viewpoints
 - Extract interaction relationships and connect viewpoints
 - Top-level viewpoints are most important for grasping a big picture, called "View"



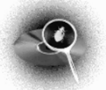
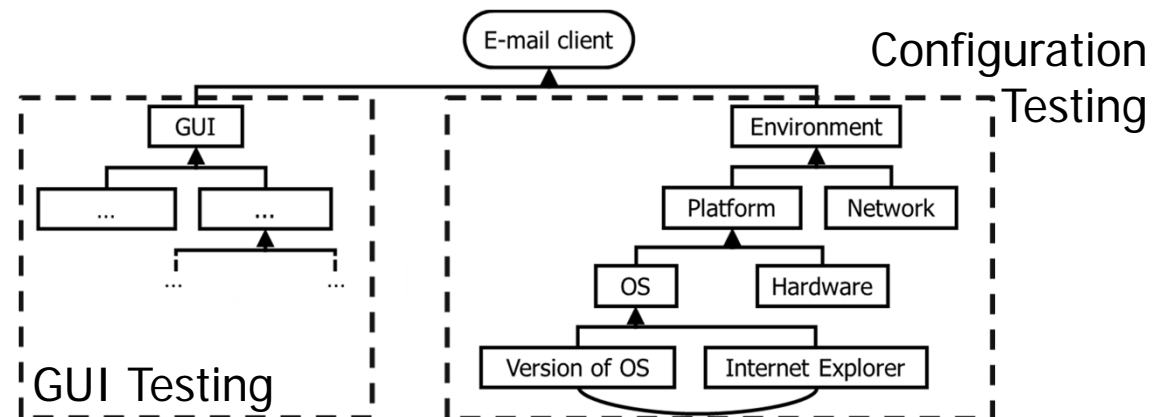
Refinement of models

- You can refine a test requirement model to make it clear and easy to understand
 - To detail viewpoints step by step to exhaust / list all test conditions
 - To move, divide or rename viewpoints if necessary
 - To check non MECE viewpoints in each layer and re-organize them as MECE
 - » MECE: Mutually Exclusive and Collectively Exhaustive
 - To check whether brotherhood viewpoints have the same stereotypes of hierarchy connections
 - To check whether interactions would be better to change viewpoints



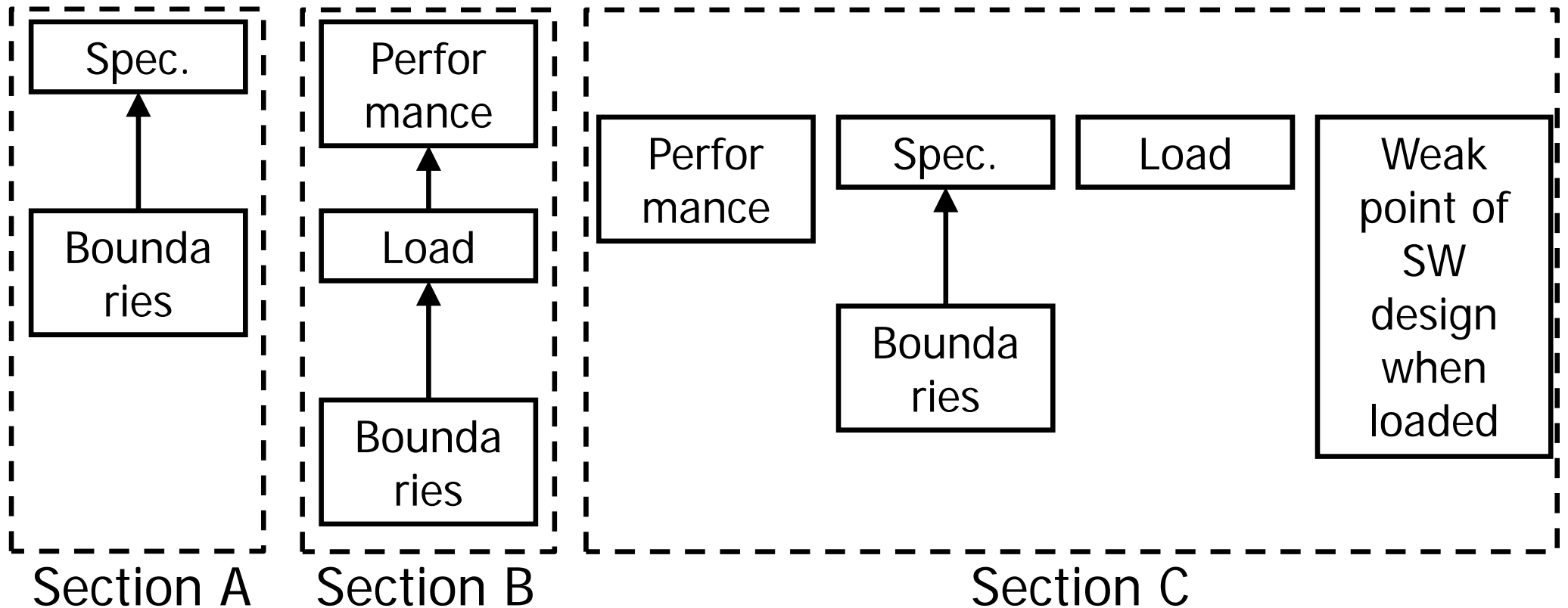
Test architecture design

- For large and complicated software and/or for multiple test levels it is necessary to make a test architecture model based on a test requirement model
 - For small and simple software and/or for a single test level test requirement model directly expresses test architecture
- Several viewpoints can be separated and organized into one large dotted-line box, “Test Container”
 - Test containers can be test levels and test types in huge test design



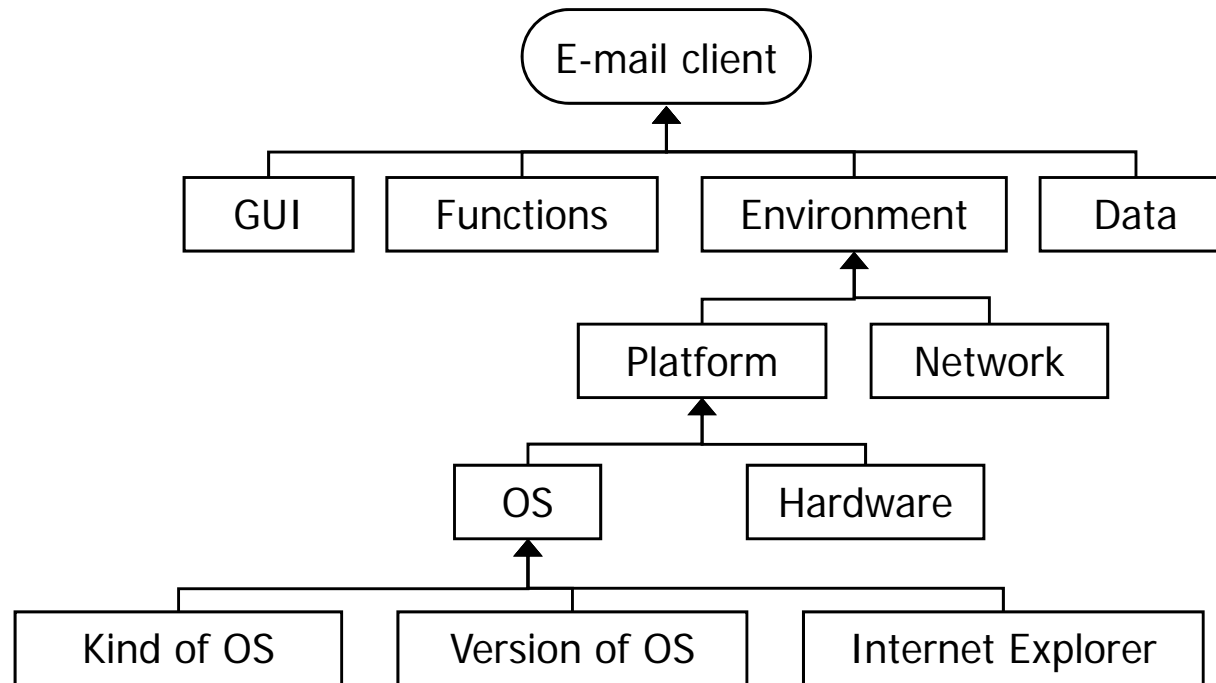
Meanings of test levels/types depend on organizations

- Viewpoints make it clear for organizations what actually to test though they use the same name of test levels/types
 - This is an example of viewpoints for performance testing by neighbor section in the same organization



Simple architecture model in this workshop

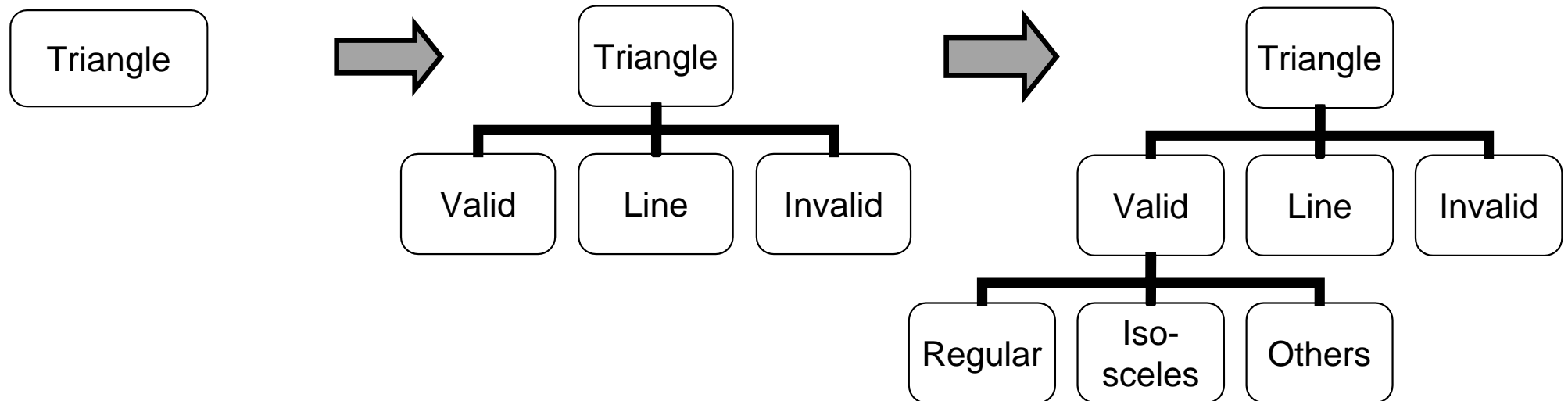
In this workshop
you can make a simple architecture model
only with viewpoints and hierarchical connections



Two approach of making test architecture

- Top-down approach

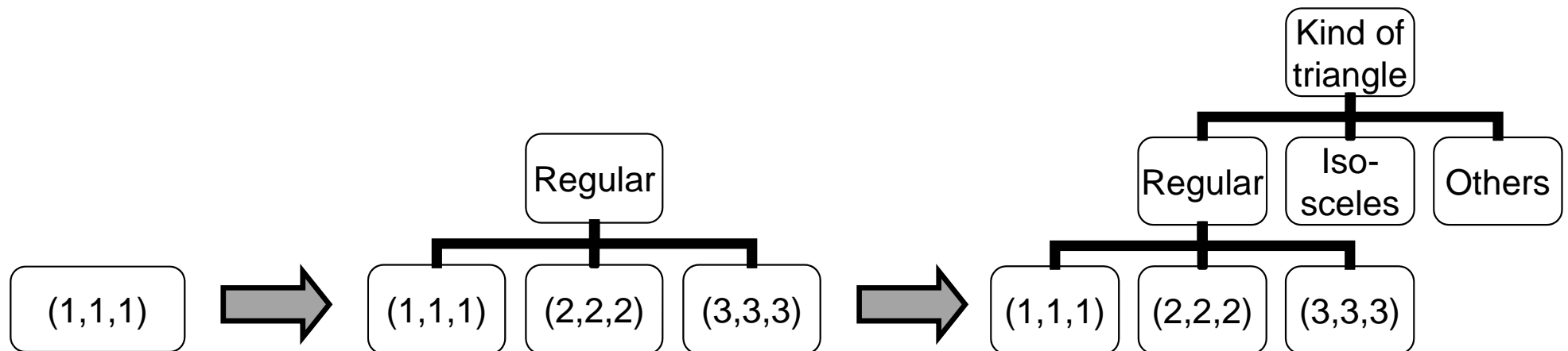
1. To write down views (top-level test viewpoints)
2. To detail viewpoints
3. To repeat above



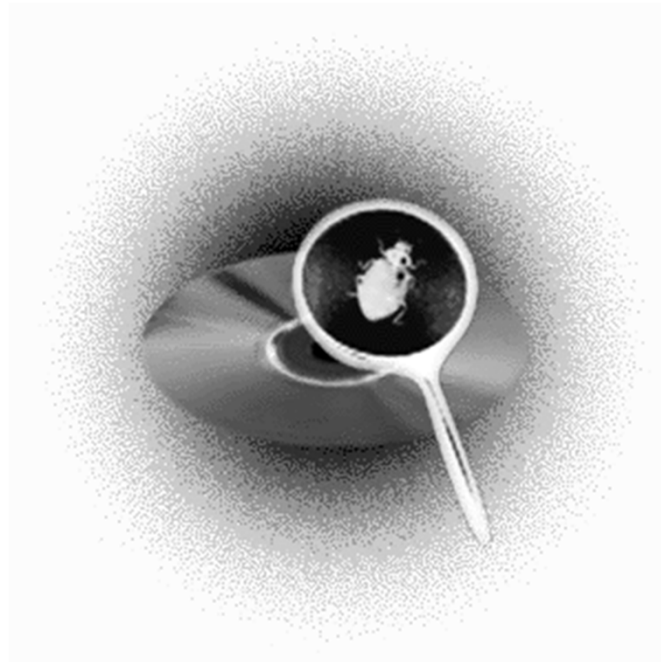
Two approach of making test architecture

- **Bottom-up approach**

1. To write down detail test cases
2. To list similar test cases or viewpoints
3. To make a group of test cases or viewpoints listed in 1. and 2.
4. To name the group and make the name a new viewpoint
5. To repeat above



Let's try to make a test architecture!



NISHI, Yasuharu
Yasuharu.Nishi@uec.ac.jp