An Introduction to Object Class Group

OMG MDA Conference on March 22, 2005

At Orlando, Florida

Dr. Peter H. Chang

Lawrence Technological University

Introduction

- **George Polya's book, *How to Solve It*, 2nd edition. Princeton University Press, 1971**
- The book **covers** numerous methods of solving problems.
- http://pup.princeton.edu/titles/669.html:

"*Generations of readers have relished Polya's deft--indeed, brilliant--instructions on stripping away irrelevancies and going straight to the heart of the problem.*"

Introduction (continued)

- Some important methods proposed by Polya in his book:

  - Analogy

  - Auxiliary Problem

  - Decomposing and recombining

  - Generalization and Specialization

  - Heuristic Reasoning

  - Signs of Progress

Analogy

Polya [1]:

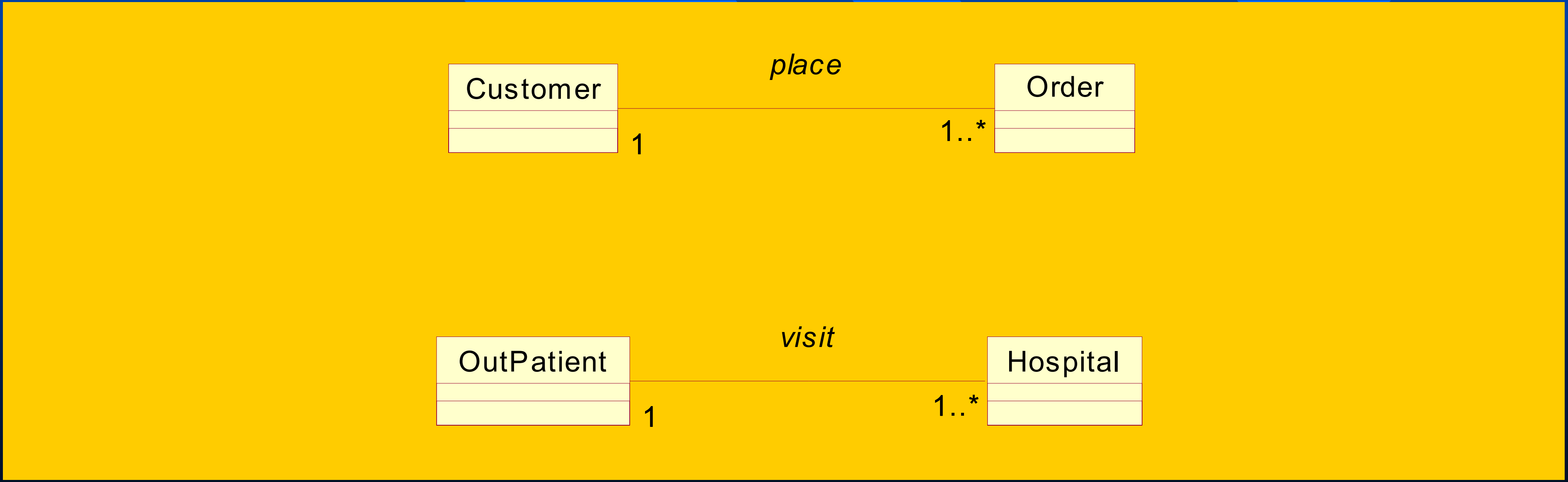
- ***Analogy*** **pervades all our thinking, our everyday speech and our trivial conclusions as well as artistic ways of expression and the highest scientific achievements. Analogy is used on very different levels. People often use vague, ambiguous, incomplete or incompletely clarified analogies, but analogy may reach the level of mathematical precision. All sorts of analogy may play a role in the discovery of the solution and so we should not neglect any sort.**

Analogous Object Class Groups

- An *object (class) group* is defined here as a group of object classes, their attributes, their associations, and other relationships such as generalization and aggregation.
- Polya [1]: Analogous *object groups* agree in certain relations of their perspective parts.
- Example 1. Object class group A: Customer, Order, and their 1 to many association. Object class group B: Outpatient, HospitalVisit, and their 1 to many association. A and B are analogous.
- Example 2. Object class group A: Airport, City, and their many to many association. Object class group B: TrainStation, City, and their many to many association. A and B are analogous.
- Example 3. Object class group A: Customer, Service, Service Provider, and two associations. Object class group B: Publisher, Book, Author and two associations.

Example of Analogous Object Class Groups

- **Customer places orders for products. Outpatients visit hospitals for treatments. The two associations *place* and *visit* shown below have the same multiplicities. Although they are not completely identical, there are some similarities. For example, payment for a customer's order includes the totaling of the prices of the products ordered; payment for an Outpatient's visit includes the totaling of the costs of the treatments that the patient received.**



Power Set

- Given a set, its power set is defined *mathematically* as the set of all its subsets, including the set itself and the empty set.
- If a set has n elements, then its power set has 2^n elements. Thus given a set S, its power set is denoted as 2^S.
- For example, given a set S = {a, b, c}, its power set is { Ø, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}}, where Ø denotes the empty set.
- Arlow and Neustatdt [2] discusses number of possible configurations for an *archetype* as 2^n if it has n optional features, and all of these features are independent variables.

Meta Power Sets

- Given a set S, we propose to define its *meta power set of nested level 1* as

$$P_1 = S \cup 2^S$$

- Inductively, for a positive integer n > 1, the *meta power set of nested level n* is defined here as

$$P_n = P_{n-1} \cup 2^{P_{n-1}}$$

- For example, for S = {a, b, c}, P_1 is

  {a, b, c, Ø, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}}

  and P_2 contains elements such as

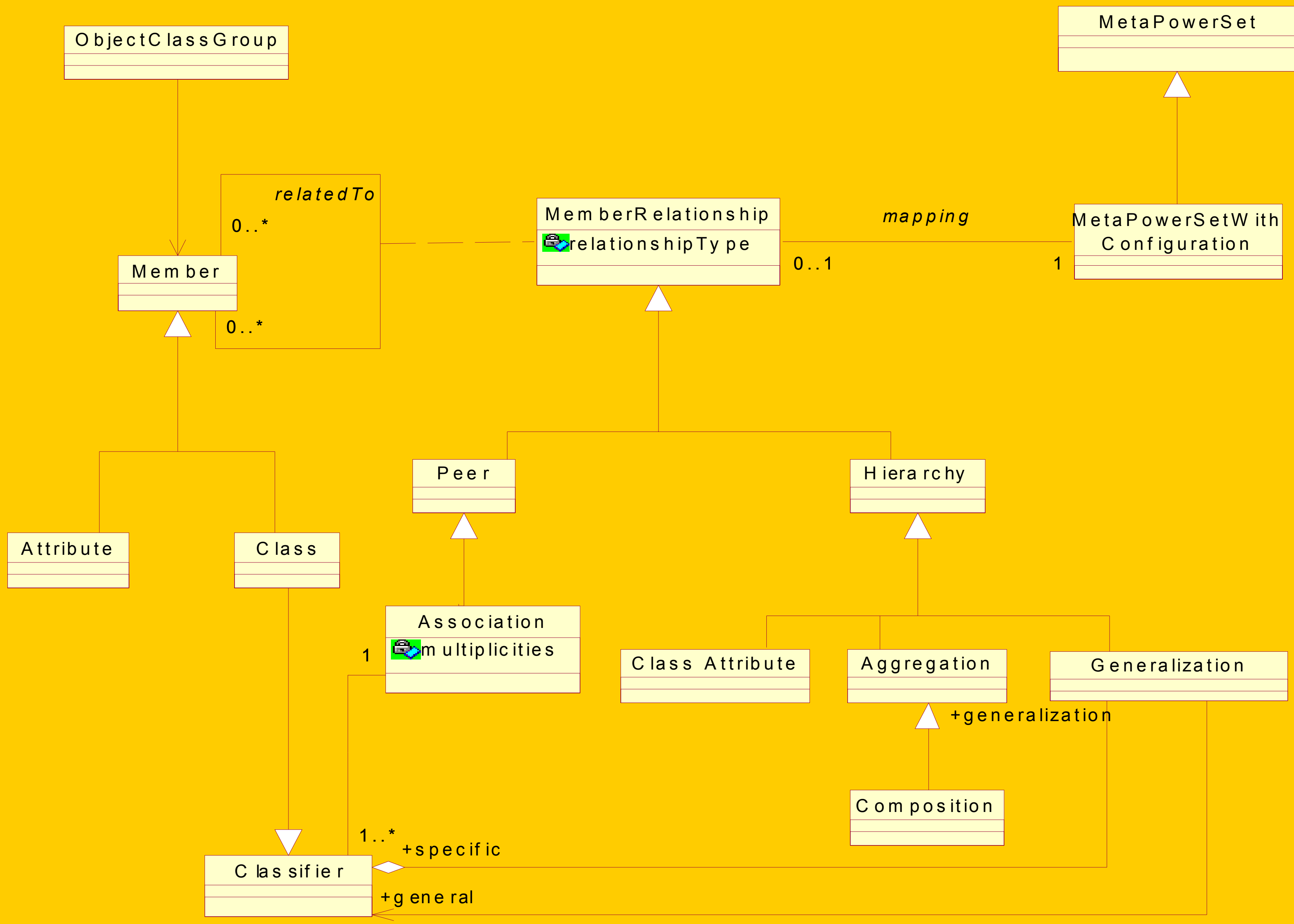
  {a, b, {c}}, {a, {b, c}}. If a and b are classes, then c may be an association between the two. If a is a class, then b and c may be subclasses of a.

Example of Object Class Group Mapped as an Element of a Meta Power Set

- Consider the object group that consists of the classes Customer, Order, and their 1 to many association.
- Let *cl_cust* denote the Customer class, *cl_ord* denote the Order class, and as_cust_ord denote their association.
- Let S denote the set {cl_cust, cl_ord, as_cust_ord}. Then {cl_cust, cl_ord, {as_cust_ord}}, an element of the power set of nested level 2, P_2, of S, describes the above object class group. We group {as_cust_ord} separately from cl_cust and cl_ord because it is an association, not a class.

A Meta Model for Object Class Groups

- In the next slide, we propose a meta model for object class groups.
- The proposed model includes a sub-model proposed by OMG in its UML specification involving with *classifier*, class, association, and generalization.
- We assume that before determining whether two object class groups are analogous, we have performed the first step of classifying their members to classes, associations, and generalizations and so on. We use the concept of meta power set to help make the determination.


ObjectClassGroup
MetaPowerSet
relatedTo
0..*
MemberRelationship
relationshipType
mapping
MetaPowerSetWith Configuration
0..1
1
Member
0..*
Peer
Hierarchy
Attribute
Class
Association
multiplicities
1
Class Attribute
Aggregation
Generalization
+generalization
Composition
1..*
+specific
Classifier
+general

An Example of the Meta Model

- Object class group A:

  classes: Customer (cl_cust), Service (cl_serv), Service Provider (cl_srvp)

  associations: customer gets services (as_get, a one to many association); service providers provide services (as_prov, a many to many association)

- Object class group B:

  classes: Publisher (cl_pub), book (cl_book), Authors (cl_auth)

  associations: publisher publishes books (as_pub, a one to many association); authors write books (as_write, a many to many association)

An Example of the Meta Model (continued)

- Object class group A mapped to Meta Power Set:
  {cl_cust, cl_serv, {as_get}}, {cl_srvp, cl_serv, {as_prov}}
- Object class group B mapped to Meta Power Set:
  {cl_pub, cl_book, {as_pub}}, {cl_auth, cl_book, {as_write}}
- A factor that is used to determine that A and B are analogous: the results of the two mappings are analogous. In addition, the multiplicities of the corresponding associations *as_get* and *as_pub* are the same and that of the corresponding associations *as_prov* and *as_write* are the same.
- In fact we can phrase the sentence that "publisher publishes books and author write books" to the sentence "publisher gets services from authors to write books".

Representation of Power Set in XML

- For a class object group containing only a Customer class that has attributes: customerName and customerNumber, it is mapped to meta power set as {cl_customer, {att_customerName, att_customerNumber}}. The mapping can be represented by the XML file in the next slide.

Representation by XML File

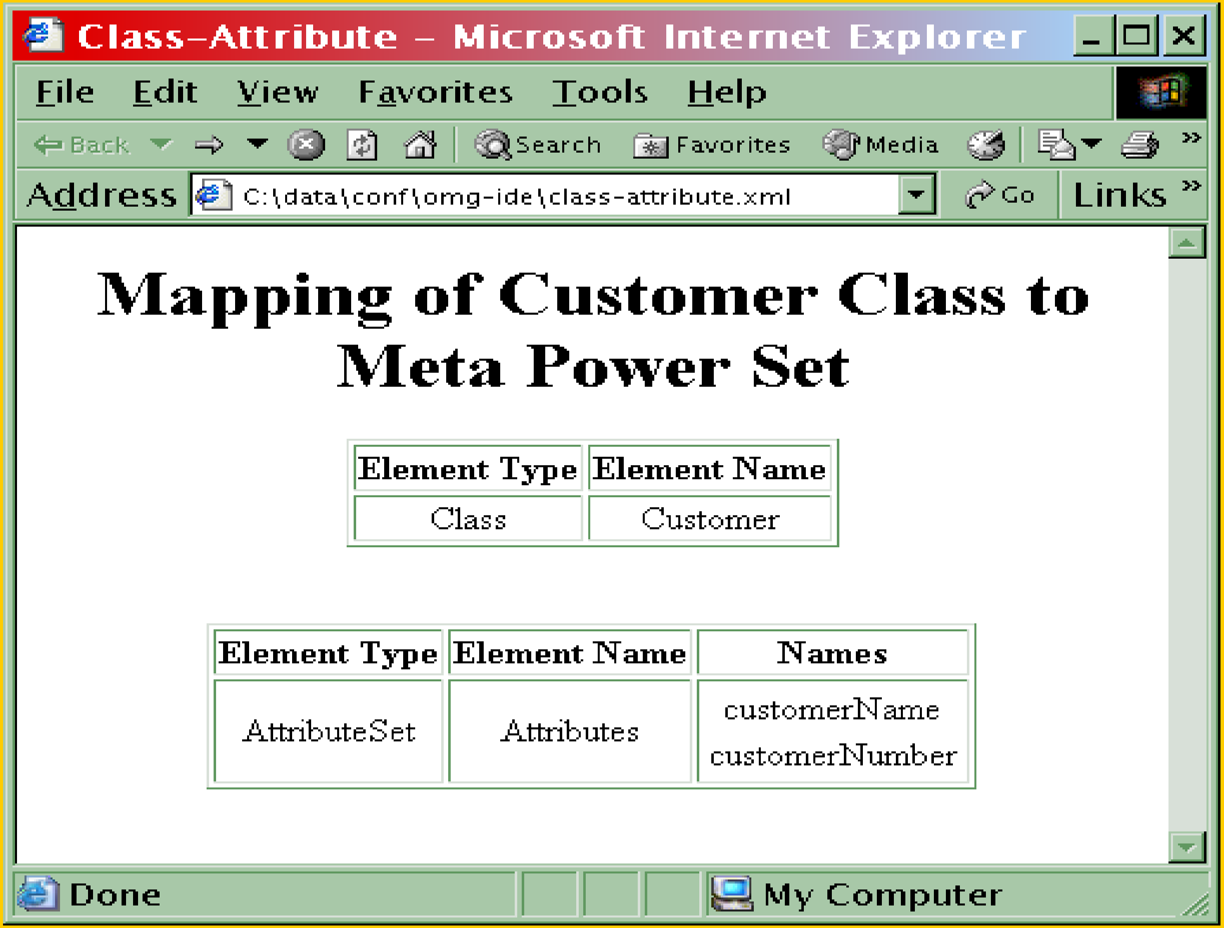
```
<?xml version = "1.0"?>

<?xml:stylesheet type = "text/xsl" href = "class-attribute.xsl"?>

<powerSetElement type = "classSet">
  <classSetElement>
    <type>Class</type>
    <name>Customer</name>
  </classSetElement>
  <attributeSet>
    <type>AttributeSet</type>
    <name>Attributes</name>
    <attributeSetElement>
      <attributeName>customerName</attributeName>
    </attributeSetElement>
    <attributeSetElement>
      <attributeName>customerNumber</attributeName>
    </attributeSetElement>
  </attributeSet>
</powerSetElement>
```

Describing Meta Power Set on a Web Page

- The last XML File may be transformed to a web page described below through rules specified in an XSL file.



Class-Attribute - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\data\conf\omg-ide\class-attribute.xml
Go Links
Done
My Computer

Mapping of Customer Class to Meta Power Set

Element Type	Element Name
Class	Customer

Element Type	Element Name	Names
AttributeSet	Attributes	customerName customerNumber

A Part of XSL File that Transforms the XML

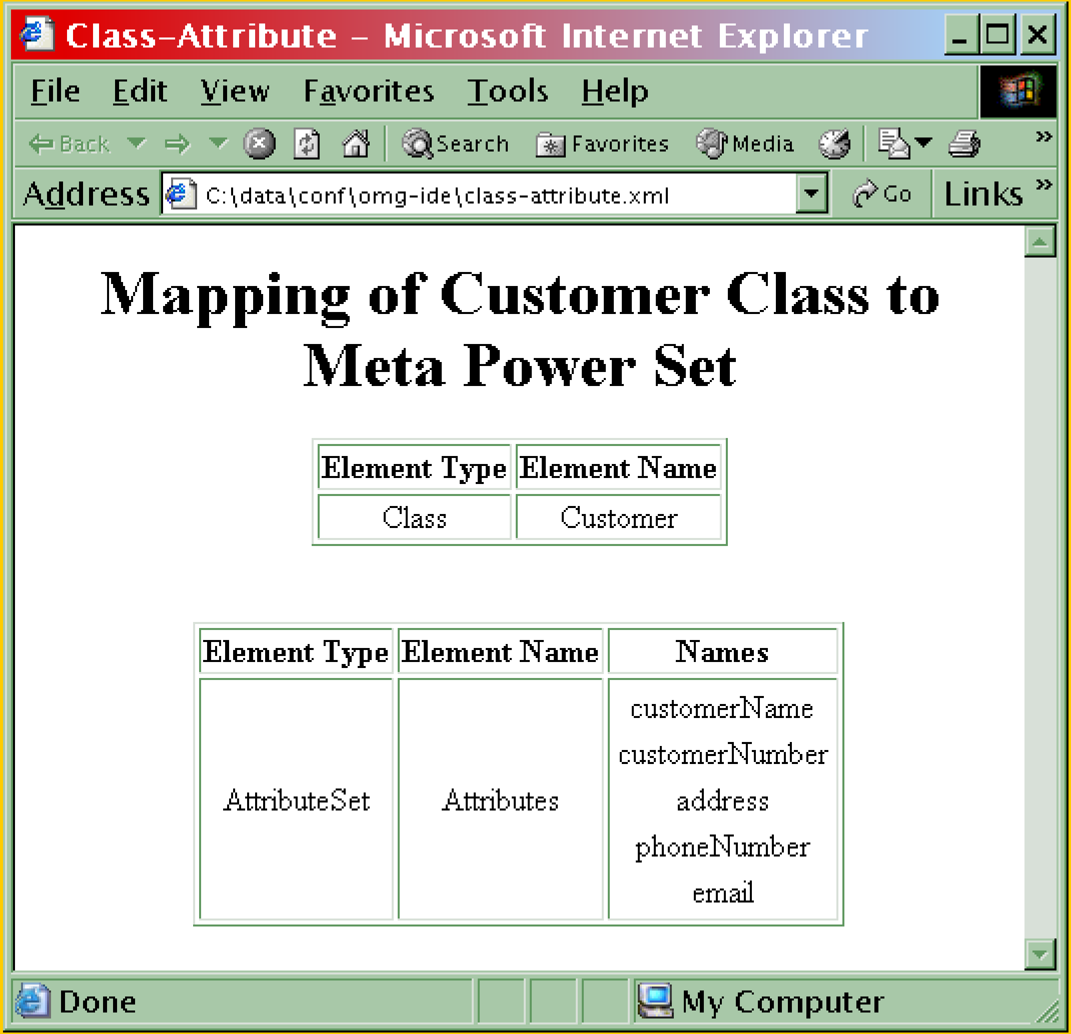
```
<table border="1">

    <!-- element xsl:text inserts text -->
    <tr><th>Element Type</th><th>Element Name</th></tr>
    <tr>
      <td><xsl:value-of select = "classSetElement/type" /></td>
      <td colspan="2"><xsl:value-of select = "classSetElement/name"
     /></td>
    </tr>
 </table>
 <br /><br />
 <table border="1">
    <tr><th>Element Type</th><th>Element Name</th>
        <th>Names</th></tr>
    <tr>
         <td><xsl:value-of select = "attributeSet/type" /></td>
         <td><xsl:value-of select = "attributeSet/name" /></td>
         <td>
         <table border ="0">
             <xsl:for-each select = "attributeSet/attributeSetElement">
               <tr><td><xsl:value-of select = "attributeName" /></td></tr>
              </xsl:for-each>
         </table>
         </td>
     </tr>

  </table>
```

Adding Attributes

- Attributes may be added to the last XML File. It can be transformed to a web page described below through the *same* XSL file.


Class-Attribute - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\data\conf\omg-ide\class-attribute.xml
Go Links
Done
My Computer

Mapping of Customer Class to Meta Power Set

Element Type	Element Name
Class	Customer

Element Type	Element Name	Names
AttributeSet	Attributes	customerName customerNumber address phoneNumber email

An Example of the Meta Model with Relationship Set

- Object class group A:

  classes: Customer (cl_cust) with attributes cust*Name* and *custId*, Service (cl_serv) with attribute *serviceType* , Service Provider (cl_srvp) with attributes *providerName* and *providerId*

  associations: customer gets services (as_get, a one to many association); service providers provide services (as_prov, a many to many association)
- Relationship set: R = {Association, Class, Attribute}
- S = A $\cup$ R. Then the meta power set with nested level 2, P_2 of S contains the following elements, among others:

  {Class, cl_cust, {Attribute, custName, custId}},

  {Class, cl_serv, {Attribute, serviceType}},

  {Association, as_get, {Class, cl_cust, cl_serv}},

  {Association, as_prov, {Class, cl_srvp, cl_serv}}

Auxiliary Problems

- Polya [1] provides a general description of auxiliary problem: *auxiliary problem is a problem which we consider, not for its own sake, but because we hope that its consideration may help us to solve another problem, our original problem. The original problem is the end we wish to attain*
- If two object class groups are analogous, then an auxiliary problem for a problem for one group may be located from problems of the other group.
- For example, in the last example, a problem of drawing up a contract between a customer and a service provider may be considered as an auxiliary problem of drawing up a contract between a publisher and the author(s) of a book.

Archetype and Heuristic Reasoning

- Jung [3]: *archetypes* arise from a common fund of human experiences (the collective unconscious) that uses archetypes as one of its ordering and structuring principles.
- Polya [1]: *Heuristic*, as an adjective, means "serving to discover". *Heuristic reasoning* is reasoning not regarded as final and strict but as provisional and plausible only, whose purpose is to discover the solution of the present problem.

  …

  Heuristic reasoning is often based on induction or on analogy.

Iterative Steps of Heuristic Reasoning

- Chang [4]: Some problems are too hard to be solved in one step. *Heuristic reasoning* and *plausible guess* are used to find an intermediate step, which is either a *partial solution* or a *new problem*. If an intermediate step is an *identified problem*, then *heuristic reasoning* and *plausible guess* are used to find a solution of this intermediate step, which is a *partial solution* of the whole problem. Iterative discovery of *signs of progress*, which is another one of Polya's methods, directs us to the *iterative and heuristic reasoning* of the next intermediate steps. This process iterates until a formal solution is found at the last step or until a determination is made regarding to that the sequence of heuristic reasoning yields no formal solution. In the latter case, an attempt for another sequence of heuristic reasoning may be made.

MDA – Abstraction and Support for Iterative and Incremental Development

- **Mellor and Balcer [5] describes the raising the level of abstraction as a foundation for MDA.**
- **MDA has 4 layers of abstraction: M0 (instance), M1 (user model), M2 (UML), M3 (MOF)**
- **MDA must support iterative and incremental development.**
- **Mappings can be developed to map a platform--independent model (PIM) to platform-specific models (PSMs).**

Meta Power Set with Configuration

- The higher level of abstraction the more object instances there may be.
- Meta power sets with various nested levels describes all possible combinations of the members of object class groups.
- As was shown earlier in the slide for meta model for object class group, *meta power set with configuration* describe the combinations of members of the object class group that were classified through a modeling process. This approach and the approach of *Iterative Steps of Heuristic Reasoning* that we proposed earlier may aid the development process of MDA.

What is next?

- ***DNA* of object class groups**

  - Define configurations

  - Determine object class groups that are analogous

  - Find auxiliary problems

  etc.

- **Iterative steps of heuristic reasoning**

  - Not all problems may be solved in one (or a few) step(s)

  - An intermediate step may be identifying an intermediate problem to solve

  - Finding an intermediate problem to solve may be as important as solving an intermediate problem

Q&A

Thank You!

References

- [1] Polya, G. *How to Solve It*, 2nd edition. Princeton University Press, 1971
- [2] Arlow J., Neustadt I. *Enterprise Patterns and MDA: building better software with archetype patterns and UML*, Addison-Wesley, 2003
- [3] Jung, C. *The Archetypes and the Collective Unconscious*, Princeton University Press, 1981
- [4] Chang, P. H. *Applying Polya's Method to Research Methodology*, Seminar Notes for the Research Methods course of the Doctor of Management Information Technology (DMIT) program at Lawrence Technological University, 2004
- [5] Mellor, S., Balcer, M. Executable UML: *A Foundation for Model-Driven Architecture*, Addison-Wesley, 2002