PROFILE-BASED COMPONENT PATTERN

[1] A. Meiappane, [2] Dr. V. Prasanna Venkatesan

[1] Research Scholar, [2] Associate Professor, Pondicherry University, Puducherry

[1] auromei@yahoo.com, [2] prasanna_v@yahoo.comm

Abstract:

The Profile-Based Component Pattern is a pattern which is used as a reusable component, when there is a need for selection of various services as per the need of the customers who were classified according to their ages as customer profile. All the services have been initiated and activated when the user needs even one service in that system, which makes the system performance to degrade. To avoid this customer has to be classified according to their ages and their usage of services are also identified. It has been observed that the maximum number customers require only minimum number of services at all time. So it’s unnecessary to initiate and activate all the services to their customers, which degrades the non-functional quality attributes like performance, reusability, maintainability and modifiability of the system. Here an example of internet banking application has been taken to show the use of profile based component pattern. This profile based component pattern selects the service as the user logs in. This doesn’t limit the user to the use of other services; indeed the user can enjoy all the services, as when there is a need for other services.

Keyword: Profile based, component, reusability, performance, maintainability, modifiability.

1. INTRODUCTION:

The components which is available in any system, is activated for the user to use in the system, which reduces the performance (Zhu Zemin , 2009). This can also be eradicated by using the design patterns, in design to improve the quality (Bass et al, 2003) . The following figure reveals the workflow design for flexible internet banking.

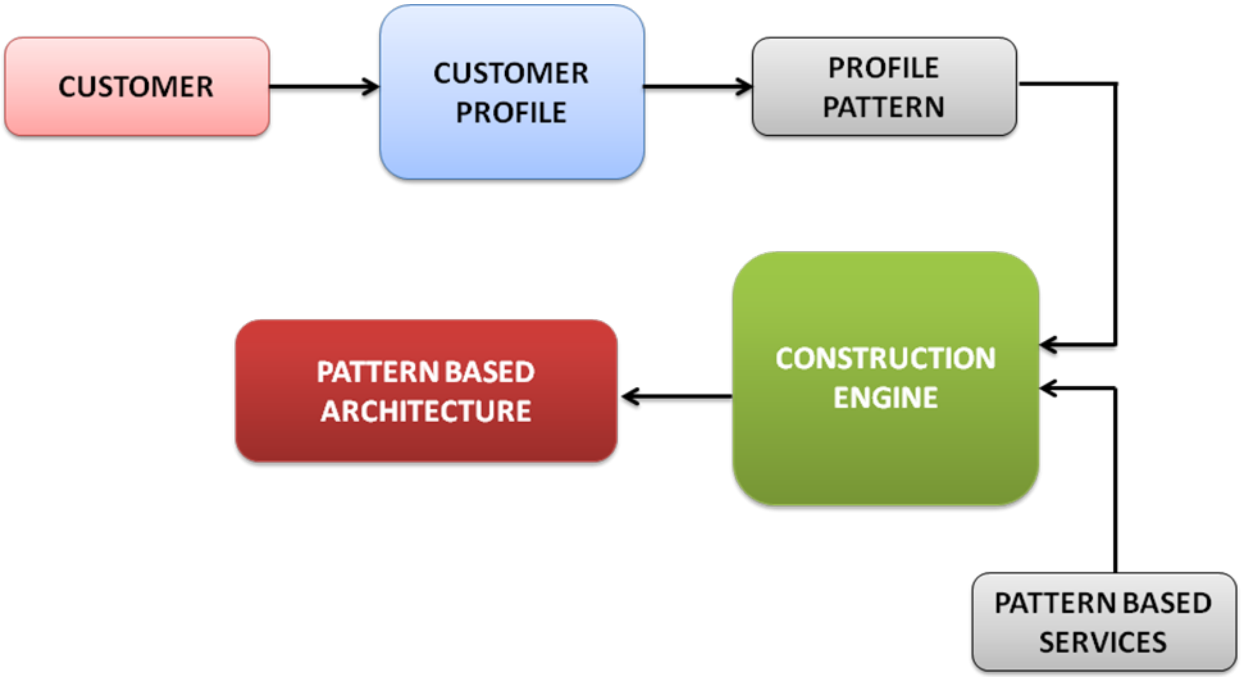

Figure 1.1 Workflow for selection of Pattern Based services

The profile-based component selects the needed component for customer. This needed component is selected on the categorization of various customer profiles. The various customers have been classified according to their ages and also the services needed by them (Meiappane et al, 2011). There is no profile-based component readily available for use (Xu Zhang et al, 2010) from the perspective of customers to be used by the developer. This component analyzes or works out the strategy for the customers, and with that profile the needed services alone activated. This intended to develop a pattern for the profile-based component. The following brings the clarity on the use of the proposed profile-based component pattern with its various uses.

2. PROFILE-BASED COMPONENT PATTERN:

NAME:

Profile-based component

CONTEXT:

In any internet-based system, all the services have been activated for the users when they log in. However, the system performance can be enhanced, when only necessary services are activated or initialized to user. It has to be determined what services should be activated for which user. The users are categorized, for example, on age profile and according to their usage of services.

PROBLEM:

How to select what services to activate?

FORCES:

1. All services for each user could be activated and initialized at once, but, then the system performance degrades.
2. A user would be provided with all services but the user needs only some
3. The computational cost of all the services provided to the user is high but the user uses only the needed service, which costs less
4. The number of hits for a service is increasing but then the network bandwidth cannot be increased

SOLUTION:

This pattern is used to construct the internet banking system. It typically consists of user authentication, access to account, Payment, registration, enquiry of details, notification. The profile-based component pattern, selects the service as the user logs in. The services provided to the user are selected according to user's profile. User's profile might contain age, history of service usage, and other characteristics. This does not limit the user for not using the other services. They can also use all other services needed for them.

EXAMPLE

In this example of the profile-based component pattern, the various components have been selected as per the customer preference. The profile-based class builds the component, for the customer, to improve the quality.

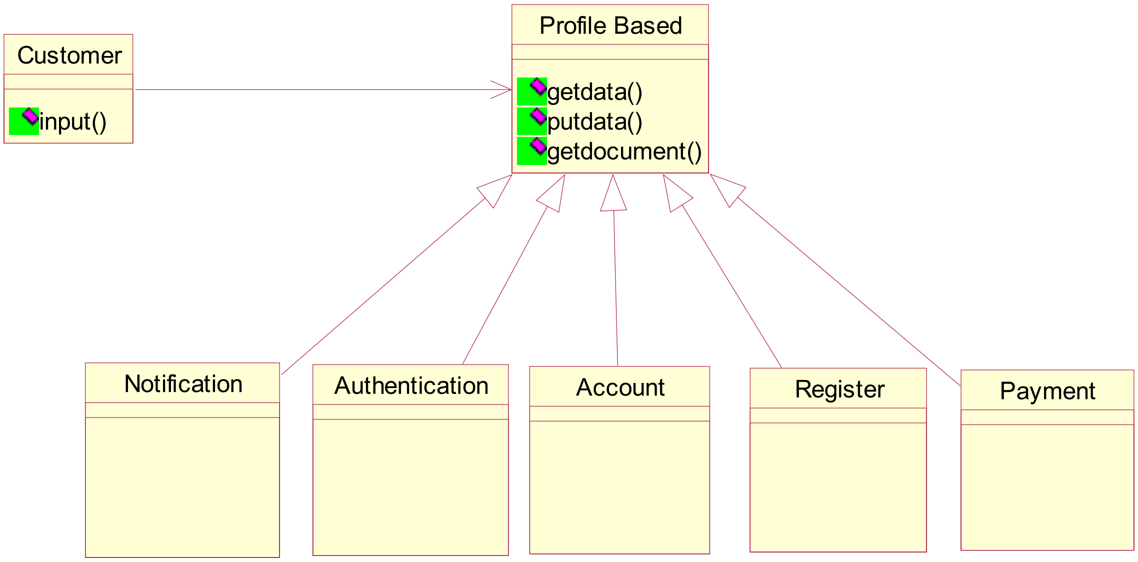

Figure 2.2 Class Diagram for Profile-Based Component Pattern

Sample code:

```
Class InternetBanking{
  private String login = "";
  private String service = "";
  private String logout = "";
  public void setlogin(String login)     { this.login = login; }
  public void setservice(String service) { this.service = service; }
  public void setlogout(String logout)             { this.logout = logout; }
}
  abstract class ProfileBased {
  protected IB ib;
  public IB getib() { return ib; }
  public void createNewcomponent() { ib = new IB(); }
  public abstract void buildlogin();
  public abstract void buildservice();
  public abstract void buildlogout();
}
class authentication extends ProfileBased {
  public void buildlogin()   { ib.setlogin("id"); }
  public void buildservice() { ib.setservice("services"); }
  public void buildlogout()  { ib.setlogout("validated"); }
}
class payment extends ProfileBased {
  public void buildlogin()   { ib.setlogin("id"); }
  public void buildservice() { ib.setservice("services"); }
  public void buildlogout()  { ib.setlogout("Cheque+card"); }
}

class InternetBankingSystem {
  private ProfileBased componentBuilder;
  public void setComponentBuilder(ComponentBuilder cb) { ComponentBuilder = cb; }
  public IB getib() { return componentBuilder.getIB(); }
  public void constructIB() {
    componentBuilder.createNewIBProduct();
    componentBuilder.buildlogin();
    componentBuilder.buildservice();
    componentBuilder.buildlogout();
  }
}
Class customer
{-----
-------
}
class Builder {
  public static void main(String[] args) {
    IBSystem ibsystem = new IBSystem();
    ProfileBased authenticate = new authenticate();
    ProfileBased payment = new payment();
    ibsystem.set ProfileBased ( authenticate );
    ibsystem.constructIB();
    IB ib = ibsystem.getIB();
  }
}
```

The class builder is used to build the profile-based component system. The internet banking system is class which constructs the component builder.

CONSEQUENCES

The profile-based component pattern is most suitable if it uses the profile of users. This pattern works well when system gets the user profile and with that data the profiles has to be categorized. The profile-based component pattern does not solve how the categorization of the user is implemented. The classification and categorization of users is another pattern.

RELATED PATTERNS:

Builder pattern

3. SUMMARY

The pattern is presented using a consistent format with the sections like name, problem, intent, context, solution, known uses, sample code, motivation, structure and participants. The patterns are catalogued by using Eric Gamma approach. Hence to improve the software quality it is necessary to catalogue and present the pattern by means of informal text and graphic diagrams.

4. ACKNOWLEDGEMENT

I A.Meiappane, want to thank sincerely, Pavel Hurby for his sincere encouragement and support during the shepherding phase. I also want to thank my research guide who is even the co-author of this paper, Dr. V. Prasanna Venkataesan for his motivation and continuous support.

5. REFERENCES

(Meiappane et al, 2011)

A. Meiappane, K. Gideon and Dr. V. Prasanna Venkatesan," Adaptive Framework of the Internet Banking Services based on Customer Classification" , *International Conference on Advances in Engg and Tech*, (ICAET-2011).

(Bass et al, 2003)

Bass, L., Clements, P. and Kazman, R. Software Architecture in Practice Addison-Wesley, 2003.

(Zhu Zemin , 2009)

Zhu Zemin,"Study and Application of Patterns in Software Reuse" , IEEE, 2009.

(Xu Zhang et al, 2010)

Xu Zhang and Chung-Horng Lung "Improving Software Performance and Reliability with an Architecture-based Self-Adaptive Framework", 34th Annual IEEE Computer Software and Application Conference, 2010