

Aus-Air Online Booking System Presentation

Team Hy-Fly:

Benny Chen - Edward Li - Kim Pham - Balint Seeber

Introduction

- Team Members
 - Balint – Team Leader
- Overview: Structured Approach + OOP
- Special Objectives – Our Claims
 - Realised the objectives to develop on-line system
 - FPB – Fast Prototyping with B

Requirement Elicitation

- Development of Proposal
 - Review existing systems
 - Initial requirements
 - Employing ISO-9126 model to review the proposal
- Final Requirements

Preliminary Design

- Drawing Out Basic Entities and Relationships
 - Entity Relationship Diagram
 - Data Flow Diagram
- Production of B - Specification

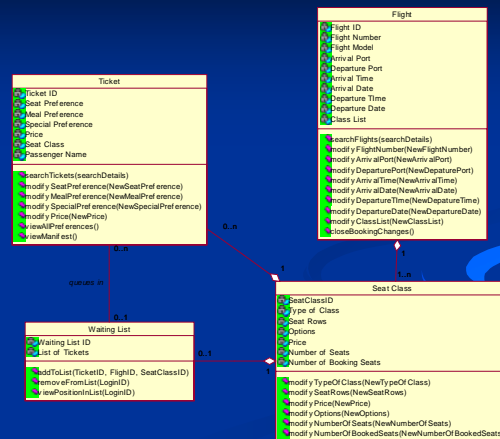
Object Oriented Design

- Use of UML
 - Class diagram
 - Use case diagram – Example: Booking
 - Activity diagram and sequence diagram
- Design Patterns
 - Example: Module-View-Control
 - Used for overall system design

Requirement Tracing

- Demonstrating Example – Waiting List
 - Requirement
 - Use case diagram
 - Specification
 - Class diagram
 - Prototype

Class diagram (to be deleted)



Prototype Demonstration

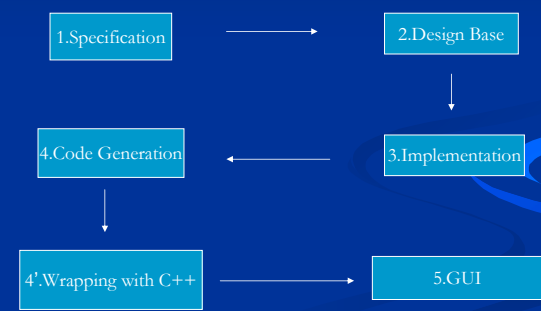
- Full Demonstration of a Registration → Booking → Cancellation Process
- System Architecture

From Design to Implementation

- Fast Prototyping with B (FPB)
 - Why "Fast Prototyping"
 - What is it?
 - Advantages and disadvantages
 - Result of the new methodology
- Wrapping B in C++
 - Maintaining object oriented
 - Advantages of wrapper
- Demo of Flight Management Across the Network

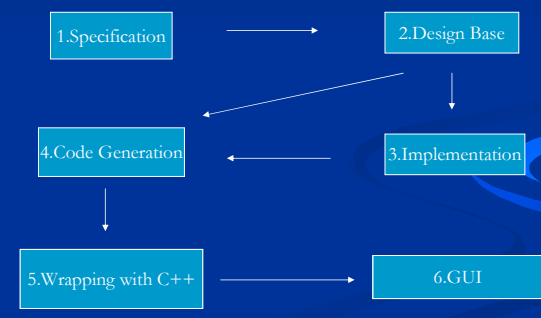
Fast Prototyping with B

Traditional



Fast Prototyping with B (cont)

■ New methodology



Wrapper

■ Fast Prototyping with B (FPB)

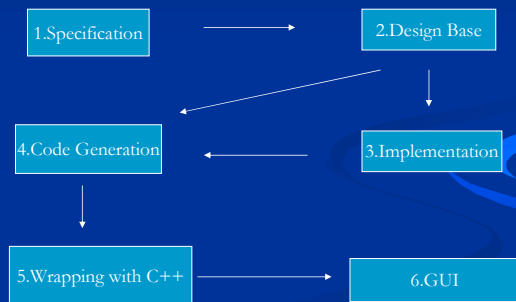
```

void CUser::addCustomerAPI () {
    nt rep, base;
    make_UserBase(&rep, &base, blah, blah, blah);
    mod_userID(userBaseID, userID);
    mod_userLogin(&rep, userBaseID,
        toString(userLogin));
    mod_userPassword(&rep, userBaseID,
        toString(userPassword));
    mod_userFirstName(&rep, userBaseID,
        toString(userFirstName));
    mod_userLastName(&rep, userBaseID,
        toString(userLastName));
    mod_userAddress(&rep, userBaseID,
        toString(userAddress));
    mod_userPhoneNo(userBaseID,
        userPhoneNumber);
    mod_userEmail(&rep, userBaseID,
        toString(userEmail));
    mod_userStatus(userBaseID,
        userLoginStatus);
}

resp <- addCustomerAPI(login_name, password,
first_name, last_name, address, phone, mobile, email) =
VAR bb, db, nm IN
nm <- nbr_CustomerRecord;
IF nm >= maxCustomer THEN
    resp := customer_database_full
ELSE
    bb, db <- key_search_customerID(login_name);
    IF bb = TRUE THEN
        resp := exist_customer
    ELSE
        resp := OK;
        bb, db <- make_UserBase(FALSE)
        ; bb <- mod_userLogin(db, login_name)
        ; bb <- mod_userPassword(db, password)
        ; bb <- mod_userFirstName(db, first_name)
        ; bb <- mod_userLastName(db, last_name)
        ; bb <- mod_userAddress(db, address)
        ; bb <- mod_userPhone(db, phone)
        ; bb <- mod_userMobile(db, mobile)
        ; bb <- mod_userEmail(db, email)
        ; print_BaseObj_CustomerRecord(db)
    END
END
END;
    
```

Future Work

- New methodology (Stage 2)



Project Management

- Time and Schedule Management
 - Gantt Chart
 - Work Breakdown Structure
 - Project Coordinator to arrange tasks
- Quality Assurance
 - Regular group meeting
 - Repository for all products
 - Peer Review

Project Management

- Risk Management
 - Story of Balint's bus accident
- Documentation
 - Requirement tracing
 - Version control
 - Release note for each version

Project Reflection

- What We Have Achieved
 - A working prototype that realised the requirements and design
 - Portfolio of documents detailing the development
 - Reconciling B with OO (different paradigms)
 - Efficiently working in a group

Project Reflection

- What We Have Learnt
 - Skills essential to Software Engineering
 - Better understanding of Software Engineering
 - Not just coding
 - Requirement is the main focus
 - Experience in developing a large real-world system
 - Time management
 - Team work

Project Reflection

- Features and methodologies we have abandoned
 - Example: the locking-and-queuing feature
- What could be done if given more time
 - Use of more design patterns
 - To-do list for the prototype

Fin.

Thank you very much for your time and patience!

Questions?

If you are interested in test-driving the system or distributing the system commercially please contact the project team on [balint \[at\] spench \[dot\] net](mailto:balint@spench.net).