OBJECT ORIENTED PROGRAMMING IN A RELATIONAL ENVIRONMENT

Software Development '90

David McGoveran

January 10, 1990

OOP IN A RELATIONAL ENVIRONMENT

- A BRIEF REVIEW OF SETS
- OBJECT ORIENTED TERMINOLOGY
- OBJECT ORIENTED PRINCIPLES
- RELATIONAL TERMINOLOGY
- THE RELATIONAL MODEL
- IMPEDANCE MISMATCH
- DO'S AND DON'TS
- AN EXAMPLE

OOP IN A RELATIONAL ENVIRONMENT

A BRIEF REVIEW OF SET THEORY I

- SETS

  A SET IS A COLLECTION OF OBJECTS EACH OF WHICH SATISFY A GIVEN CONDITION AND EACH OF WHICH IS UNIQUE.

- SET PROPERTIES

  THOSE ASPECTS OF A COLLECTION WHICH SERVE TO SATISFY (OR DENY) THE DEFINING CONDITION OF A SET.

- SET MEMBERSHIP

  EACH OF THE OBJECTS IN THE COLLECTION IS SAID TO BE A "MEMBER" OF THE SET.

- SUBSET

  A SUBSET OF A GIVEN SET IS A COLLECTION OF OBJECTS, EACH OF WHICH IS A MEMBER OF THE SET.

A BRIEF REVIEW OF SET THEORY II

- SUPERSET

  A SUPERSET OF A GIVEN SET IS A COLLECTION OF OBJECTS SUCH THAT EACH MEMBER OF THE SET IS ALSO A MEMBER OF THE SUPERSET.

- UNION

  THE UNION OF TWO SETS IS THE SET WHOSE MEMBERS ARE ALSO MEMBERS OF AT LEAST ONE OF THE TWO SETS. NOTE THAT IF A MEMBER OCCURS IN BOTH SETS, IT OCCURS ONLY ONCE IN THE UNION.

- INTERSECTION

  THE INTERSECTION OF TWO SETS IS THE SET WHOSE MEMBERS ARE ALSO MEMBERS OF BOTH OF THE TWO SETS.

OOP IN A RELATIONAL ENVIRONMENT

A BRIEF REVIEW OF SET THEORY III

- DIFFERENCE

  THE DIFFERENCE OF TWO SETS (A-B) IS THE SET WHOSE MEMBERS ARE MEMBERS OF ONE SET (A) BUT NOT OF THE OTHER (B). NOTE THAT THE SET DIFFERENCE A-B IS IN GENERAL DIFFERENT FROM B-A.

- CARTESIAN PRODUCT

  THE CARTESIAN PRODUCT IS THE SET OF ALL POSSIBLE PAIRS OF MEMBERS, ONE FROM EACH OF THE TWO SETS.

OOP IN A RELATIONAL ENVIRONMENT

OBJECT ORIENTED TERMINOLOGY I

- ABSTRACT DATA TYPE

  AN ABSTRACT DATA TYPE IS A SET OF SERVICES AND PROPERTIES.

- CLASS

  A CLASS IS AN IMPLEMENTATION OF AN ABSTRACT DATA TYPE.

- ENCAPSULATION

  HIDING THE WAY IN WHICH BOTH PROPERTIES AND SERVICES ARE IMPLEMENTED.

- OBJECT

  AN OBJECT IS AN ENCAPSULATION OF AN INSTANCE OF SOME PARTICULAR CLASS.

- INSTANCE

  AN INSTANCE OF A CLASS IS AN OBJECT WHOSE PROPERTIES HAVE PARTICULAR VALUES.

OBJECT ORIENTED TERMINOLOGY II

- FEATURES

  A CLASS IS CHARACTERIZED BY ITS FEATURES: ATTRIBUTES AND ROUTINES.

  - ATTRIBUTES

    AN ATTRIBUTE IS A COMPONENT OF A CLASS WHICH CORRESPONDS TO A FIELD IN AN OBJECT OF THE CLASS.

  - ROUTINES

    A ROUTINE IS A COMPONENT OF A CLASS WHICH CORRESPONDS TO A METHOD IN AN OBJECT OF THE CLASS. ROUTINES ARE EITHER FUNCTIONS OR PROCEDURES.

- METHOD

  A METHOD IS A SERVICE PROVIDED BY AN OBJECT WHICH MANIPULATES THE OBJECTS PROPERTIES.

OOP IN A RELATIONAL ENVIRONMENT

OBJECT ORIENTED TERMINOLOGY III

- STATE

  THE STATE OF AN OBJECT INSTANCE IS REPRESENTED BY THE PARTICULAR VALUES OF ITS PROPERTIES. METHODS MAY CHANGE THE STATE OF AN OBJECT.

- REPRESENTATION INDEPENDENCE

  REPRESENTATION CHANGES OF AN OBJECT SHOULD REMAIN INVISIBLE TO THE CLIENT (DURING THE PROJECT LIFE-CYCLE AND DURING EXECUTION).

- DYNAMIC BINDING (SUBCONTRACTING)

  THE PROCESS OF ESTABLISHING OPERATORS AND OPERANDS ADAPTED TO THE CORRESPONDING INSTANCE AT RUN-TIME.

- PACKAGES

  AN IMPLEMENTATION OF THE DATA STRUCTURES TO WHICH A SET OF OPERATIONS APPLY.

OOP IN A RELATIONAL ENVIRONMENT

OBJECT ORIENTED TERMINOLOGY IV

- OVERLOADING

  THE ABILITY TO ATTACH MORE THAN ONE MEANING TO A NAME.

- GENERICITY

  THE ABILITY TO DEFINE PARAMETERIZED MODULES.

- CONSTRUCTOR

  A PROCEDURE USED TO CREATE AN INSTANCE OF AN OBJECT.

- ACCESSOR

  A FUNCTION USED TO ACCESS THE PROPERTIES OF AN OBJECT.

OOP IN A RELATIONAL ENVIRONMENT

OBJECT ORIENTED TERMINOLOGY V

- PERSISTENT OBJECTS

  OBJECTS WHICH SURVIVE BETWEEN RUN-TIME SESSIONS.

- PASSIVATION

  A PROCEDURE USED TO BUILD A STORABLE SYMBOLIC REPRESENTATION OF AN OBJECT.

- ACTIVATION

  A FUNCTION USED TO RECONSTRUCT AN OBJECT FROM ITS STORED SYMBOLIC REPRESENTATION.

- TRANSFORMER

  A FUNCTION WHICH CREATES A NEW OBJECT FROM ONE OR MORE OBJECTS OF THE SAME CLASS.

OOP IN A RELATIONAL ENVIRONMENT

OBJECT ORIENTED TERMINOLOGY VI

- CLIENT/SUPPLIER

  A CLASS IS A CLIENT OF ANOTHER IF IT USES THAT CLASSES SERVICES. THE CLASS ITSELF IS CALLED A SUPPLIER.

- DESCENDENT/HEIR

  A CLASS IS A DESCENDENT OR HEIR OF ANOTHER CLASS IF IT IS DESIGNED AS AN EXTENSION OR SPECIALIZATION OF THE CLASS.

- INHERITANCE

  THE PROCESS BY WHICH A CLASS IS DESIGNED AS A DESCENDENT OF ANOTHER CLASS.

  - MULTIPLE
  - REPEATED
  - SELECTIVE

OBJECT ORIENTED TERMINOLOGY VII

- POLYMORPHISM

  RUN-TIME ABILITY OF AN ENTITY TO REFER TO INSTANCES OF VARIOUS CLASSES.

  - REDEFINITION OF A PARENT FEATURE
  - RENAMING OF A FEATURE
  - DYNAMIC ALIASING - ACCESS THROUGH MULTIPLE NAMES

- CONCURRENCY

  ABILITY OF MULTIPLE ENTITIES OR INSTANCES TO USE AN OBJECTS METHODS SIMULTANEOUSLY.

- UNIFORM REFERENCE

  UNIFORM DATA VALUE REFERENCE, COMPUTED OR STORED.

- INVARIANTS

  ASSERTIONS TO BE SATISFIED BY EVERY INSTANCE OF A CLASS.

OOP IN A RELATIONAL ENVIRONMENT

OBJECT ORIENTED PRINCIPLES

- ORDER

  CONTRARY TO COMMON OPINION, OOP TAKES A NEUTRAL APPROACH TOWARD ORDERING - THERE IS NO TOP TO THE CALL STRUCTURE, NO HIERARCHY, NO APPLICATION-WIDE FUNCTIONAL DECOMPOSITION.

- THIS ASSUMES A COMPLETE ORTHOGONAL BASIS

  A MINIMAL SET OF CLASSES (CALLED THE BASIS) EXIST SUCH THAT:

  - EVERY CLASS TO BE IMPLEMENTED CONSISTS SOLELY OF PROPERTIES AND SERVICES HELD BY THE UNION OF THIS SET OF CLASSES
  - EACH CLASS OF THE BASIS SHARES NO PROPERTIES OR SERVICES WITH ANY OTHER CLASS OF THE BASIS
  - NO SUPERCLASS OF ANY OF THE BASIS CLASSES WILL BE CREATED - I.E. THE CLASSES TO BE IMPLEMENTED REQUIRE ONLY MULTIPLE INHERITANCE FROM CLASSES IN THE BASIS.

OOP IN A RELATIONAL ENVIRONMENT

RELATIONAL TERMINOLOGY I

- ENTITY

  A CLASS TO BE MODELED IN THE DATABASE

- RELATIONSHIP

  A DIRECTED MAP BETWEEN THE INSTANCES OF TWO ENTITIES 1:1, 1:M, M:M

- RELATIONS ARE NOT TABLES

  A RELATION IS A SPECIAL KIND OF TABLE CONTAINING A SET OF TUPLES AND REPRESENTING A CLASS.

- TUPLES ARE NOT RECORDS

  A TUPLE IS A SET OF ATTRIBUTES REPRESENTING AN ENTITY OR RELATIONSHIP.

- ATTRIBUTES ARE NOT FIELDS

  AN ATTRIBUTE REPRESENTS A CLASS PROPERTY AND IS IMPLEMENTED AS A VALUE TAKEN FROM THE DOMAIN OF THE CLASS.

RELATIONAL TERMINOLOGY II

- DOMAINS

  A DOMAIN IS A SET OF VALUES OF AN ATOMIC DATA TYPE'S PROPERTY.

- JOIN

  A JOIN BETWEEN TWO RELATIONS IS THE CARTESIAN PRODUCT OF ALL THE TUPLES OF THE TWO RELATIONS. THERE ARE VARIOUS KINDS OF JOINS WHICH ARE MORE USEFUL AND WHICH RESTRICT THE TUPLES PARTICIPATING IN THE CARTESIAN PRODUCT. (SUPERCLASS)

- PROJECTION

  A PROJECTION OF A RELATION IS THE RELATION FORMED BY A SUBSET OF THE ATTRIBUTES OF ALL THE TUPLES IN THE RELATION (SELECTIVE INHERITANCE).

OOP IN A RELATIONAL ENVIRONMENT

RELATIONAL TERMINOLOGY III

- RESTRICTION

  A RESTRICTION OF A RELATION IS THE RELATION FORMED BY A SUBSET OF THE TUPLES IN THE RELATION (A CONSTRUCTOR OR TRANSFORMER FUNCTION).

- NULLS

  A NON-VALUE INDICATOR USED WHEN NO VALUE IS KNOWN FOR AN ATTRIBUTE OR WHEN A VALUE WOULD NOT BE APPLICABLE. THESE INDICATORS MUST BE SYSTEMATICALLY AND UNIFORMLY IMPLEMENTED IF AT ALL.

- DATA INDEPENDENCE

  THE UNIMPAIRMENT OF DATABASE MANIPULATION WHENEVER INFORMATION PRESERVING CHANGES ARE MADE TO DATABASE RELATIONS (IF POSSIBLE).

OOP IN A RELATIONAL ENVIRONMENT

RELATIONAL TERMINOLOGY IV

- TRANSACTION - A LOGICAL UNIT OF WORK.

- CONCURRENCY

  THE INTERLEAVING OF TRANSACTIONS IN SUCH A WAY AS TO MAINTAIN DATABASE INTEGRITY.

- VIEWS

  A RELATION IN WHICH THE VALUES OF THE ATTRIBUTES ARE CREATED AT RUN-TIME.

- STORED OR DATABASE PROCEDURES

  A COMPILED OR PARTIALLY INTERPRETED PROCEDURE USED TO MANIPULATE DATABASE OBJECTS AND WHICH CAN BE INVOKED INTERACTIVELY OR PROGRAMMATICALLY BY NAME AND WITH PARAMETERS.

- DATABASE TRIGGERS

  A DATABASE PROCEDURE WHICH IS INVOKED AUTOMATICALLY WHENEVER A SPECIFIC EVENT OCCURS.

OOP IN A RELATIONAL ENVIRONMENT

THE RELATIONAL MODEL IN THREE PARTS I

- STRUCTURAL

  ALL INFORMATION IS REPRESENTED EXPLICITLY AT THE LOGICAL LEVEL AND ONLY BY VALUES IN RELATIONAL TABLES.

- MANIPULATIVE

  THERE IS AT LEAST ONE SET-ORIENTED DATA SUBLANGUAGE WHOSE STATEMENTS ARE EXPRESSIBLE PER SOME WELL-DEFINED SYNTAX AS CHARACTER STRINGS AND WHICH SUPPORTS ALL OF THE FOLLOWING:

  - DATA DEFINITION
  - VIEW DEFINITION
  - INTERACTIVE AND PROGRAMMATIC DATA MANIPULATION
  - INTEGRITY CONSTRAINTS
  - AUTHORIZATION
  - TRANSACTION BOUNDARIES

OOP IN A RELATIONAL ENVIRONMENT

THE RELATIONAL MODEL IN THREE PARTS II

- NON-SUBVERTIBLE INTEGRITY (PRE-CONDITIONS, POST-CONDITIONS)
  - REFERENTIAL

    IF AN ATTRIBUTE IN A RELATION REFERS TO THE PRIMARY KEY OF ANOTHER RELATION, VALUES MAY NOT EXIST FOR WHICH THERE IS NO REFERENCE.
  - ENTITY (UNIQUENESS)

    EACH AND EVERY ATOMIC VALUE IS LOGICALLY ACCESSIBLE VIA RELATION NAME, PRIMARY KEY, AND ATTRIBUTE NAME.
  - DOMAIN

    IT IS NOT POSSIBLE TO ASSIGN A VALUE TO AN ATTRIBUTE THAT DOES NOT BELONG TO THE DOMAIN OF THE ATTRIBUTE.

OOP IN A RELATIONAL ENVIRONMENT

SOME CORRESPONDENCES I

- BOTH OBJECT-ORIENTED AND RELATIONAL ARE "SET" MODELS
- OBJECTS ARE ENTITIES
- STORED PROCEDURES ARE DATABASE METHODS
- A SET OF STORED PROCEDURES CAN BE USED TO DEFINE AN ABSTRACT DATA TYPE IF THE ONLY WAY TO UPDATE/ACCESS THE DATA FOR THE TYPE IS THROUGH THE PROCEDURES
- TRIGGERS CAN BE USED TO ENFORCE PRE-CONDITIONS AND AXIOMS (POST-CONDITIONS)
- VIEWS ARE AN ATTEMPT TO IMPLEMENT UNIFORM REFERENCE

OOP IN A RELATIONAL ENVIRONMENT

SOME CORRESPONDENCES II

- A JOIN IMPLEMENTS A SUPERCLASS
- A PROJECTION IMPLEMENTS SELECTIVE INHERITANCE
- A RESTRICTION IS A CONSTRUCTOR/TRANSFORMER
- AN INTEGRITY CONSTRAINT IS AN INVARIANT
- DATABASE SECURITY CAN BE USED TO ENFORCE ENCAPSULATION
- CAREFUL USE OF NESTED PROCEDURES AND EXECUTION PERMISSIONS CAN BE USED TO SIMULATE INHERITANCE
- SUPERCLASSES MAY CONTAIN DATA WHICH ARE LOGICAL VIEWS OF THE DATABASE

OOP IN A RELATIONAL ENVIRONMENT

IMPEDANCE MISMATCH: DEF. & KEY FACTORS I

IMPEDANCE MISMATCH IS A TERM FOR THE CONFLICT BETWEEN TWO DATA MODELS. IT AFFECTS THE EFFICIENCY OF ANY SYSTEM WHICH SEEKS TO USE BOTH DATA MODELS.

- NUMBER OF ATOMIC STEPS REQUIRED TO EFFECT A CHANGE OF STATE

- NUMBER OF ATOMIC (I.E. STRUCTURELESS) DATA VALUES REQUIRED TO RECORD A STATE
  - REDUNDANCY, ABSTRACTIONS, DYNAMIC VS. STATIC STRUCTURES

- COMPLEXITY OF STATE STRUCTURE
  - HIERARCHIES, NETWORKS, LINKED LISTS, ETC. VS. FLAT

OOP IN A RELATIONAL ENVIRONMENT

IMPEDANCE MISMATCH: DEF. & KEY FACTORS II

- NUMBER OF DATA TYPES
- NUMBER OF BASE OBJECTS OR ENTITIES WHICH HAVE A STATE
- OPTIMAL TRANSACTION LENGTH
- INDEPENDENCE OF STATES
  - LOCK SCOPE (GRANULARITY AND DURATION)
- STRUCTURAL FLEXIBILITY - SCHEMA CHANGES

OOP IN A RELATIONAL ENVIRONMENT

DON'TS: OBJECT-ORIENTED & RELATIONAL MODELS IN CONFLICT

- PERFORM UNNECESSARY PROCEDURAL UPDATES
- EXTENDED TRANSACTION PERIODS
- EXTENDED LOCK SCOPE
- DENORMALIZED (OR NEVER NORMALIZED) DATABASES
- COMPLEX OBJECT DATA STRUCTURES
- REDUNDANCY
- UPDATE IN PLACE OF NON-PERSISTENT OBJECTS
- ACTIVATE OBJECTS UNNECESSARILY

OOP IN A RELATIONAL ENVIRONMENT

DO'S: MINIMIZING THE IMPEDANCE MISMATCH I

- BUILD AN ISOLATION LAYER
- DESIGN BASE OBJECTS WITH FLAT DATA STRUCTURES
- DESIGN SUPEROBJECTS
- GIVE SUPEROBJECTS SET-AT-A-TIME METHODS
- IDENTITY BASE OBJECTS WITH DATABASE ENTITIES AND VICE-VERSA
- MAINTAIN EXPLICIT LOCKS FOR LONG TRANSACTIONS
- MINIMIZE LOCK SCOPE

OOP IN A RELATIONAL ENVIRONMENT

DO'S: MINIMIZING THE IMPEDANCE MISMATCH II

- DEFER DATABASE TRANSACTIONS
- USE ACTIVATION AND PASSIVATION TO MINIMIZE THE EFFECT OF "BROWSE AND UPDATE"
- USE STORED OR DATABASE PROCEDURES
- USE DATABASE TRIGGERS
- USE DATABASE INTEGRITY ENFORCEMENT

OOP IN A RELATIONAL ENVIRONMENT

A PORTFOLIO MANAGEMENT EXAMPLE: BAD I

- IMPLEMENTATION
  - PORTFOLIOS CONSIST OF A COLLECTION OF DATA ABOUT SECURITIES AND RELATED DATA
  - DATA IS MAINTAINED IN RECORDS, SOME OF WHICH HAVE REPEATING GROUPS
  - TRANSACTION MANAGEMENT AND RECOVERY IS LEFT TO THE DATABASE
  - STATE OF THE PORTFOLIO OBJECT IS PASSIVATED SPREADSHEET-STYLE WITH REPEATING GROUPS AS NON-ATOMIC FIELDS

OOP IN A RELATIONAL ENVIRONMENT

A PORTFOLIO MANAGEMENT EXAMPLE: BAD II

- PROBLEMS
  - RELATIONAL LANGUAGE CAN NOT BE USED TO SELECT SECURITIES FROM THE PORTFOLIO
  - INTEGRITY CAN NOT BE ENFORCED BY THE DATABASE - FOR EXAMPLE, WHEN SECURITIES ARE TRADED - WHEN TRANSFERRED FROM ONE PORTFOLIO TO THE OTHER
  - IF THE DATABASE IS NORMALIZED, THE ACTIVATION/PASSIVATION CODE BECOMES COMPLICATED
  - RELATIONAL LANGUAGE CAN NOT BE USED TO GENERATE SUMMARY REPORTS ON ALL PORTFOLIOS
  - TRANSACTION DURATION AND LOCK SCOPE IS POORLY MANAGED
  - CONCURRENCY IS DIFFICULT

OOP IN A RELATIONAL ENVIRONMENT

A PORTFOLIO MANAGEMENT EXAMPLE: GOOD I

- IMPLEMENTATION
  - PORTFOLIOS CONSISTS OF A COLLECTION OF SECURITIES
  - SECURITIES HAVE TYPES
  - DATA MODEL IS RELATIONAL AND NORMALIZED
  - DATA IN EACH BASE OBJECT REFERENCES ONLY RELATIONAL VIEWS OF BASE TABLES
  - DATA IN EACH BASE OBJECT IS FLAT AND IS HELD BY REFERENCE ONLY - I.E. THERE IS NO ATTEMPT TO PROVIDE LOCALITY OF REFERENCE AT THE BASE OBJECT LEVEL

OOP IN A RELATIONAL ENVIRONMENT

A PORTFOLIO MANAGEMENT EXAMPLE: GOOD II

- AN ISOLATION LAYER IS USED FOR ALL ACTIVATION/PASSIVATION AND ALL ON-LINE DATABASE MANIPULATION

- WHEN ANY OBJECT IS ACTIVATED, A CONTROL TABLE IS UPDATED TO REFLECT WHAT USER/PROCESS HAS "CHECKED OUT" THE DATA

- WHEN ANY OBJECT IS PASSIVATED, THE CONTROL TABLE IS CHECKED FOR CONFLICTS

- ACTIVATION AND PASSIVATION ARE PERFORMED FOR SUPEROBJECTS WHERE POSSIBLE. THIS ALLOWS AN ENTIRE PORTFOLIO TO BE LOADED WITH A FEW SQL SELECT STATEMENTS, EACH OF WHICH IS USED TO ACCESS SETS OF DATA. THIS DATA IS THEN WRITTEN TO AN IN CORE LOCATION AND EACH SECURITY CAN POINT TO THE APPROPRIATE RECORDS. AS LONG AS NEITHER THE POINTERS NOR THE IN CORE LOCATION ARE PUBLIC, ENCAPSULATION IS MAINTAINED.

OOP IN A RELATIONAL ENVIRONMENT

ISOLATION LAYER CHARACTERISTICS

- IS A DATABASE "MESSENGER"

- USES STORED PROCEDURES RATHER THAN EMBEDDED SQL

- HANDLES MULTIPLE STATEMENTS IN ONE CALL

- AUTOMATICALLY UPDATES/CHECKS THE CONTROL TABLE

- CAN TRANSLATE FLAT TABLES TO OTHER "FLAT" DATA STRUCTURES -

  I.E. EACH FIELD IN THE STRUCTURE CAN BE REFERENCED FROM A SINGLE POINTER FOR THE "RECORD", AND A TRAVERSAL/ALLOCATION ALGORITHM EXISTS WHICH TAKES A POINTER TO A "RECORD" AS INPUT AND RETURNS A POINTER TO THE NEXT "RECORD".

- CAN PERFORM LEGAL DATA TYPE CONVERSIONS