

0
3-24-97 881

SANDIA REPORT

SAND97-8208 • UC-405

Unlimited Release

Printed November 1996

SAND--97-8208

M97052193

Kaiser Permanente - Sandia National Health Care Model

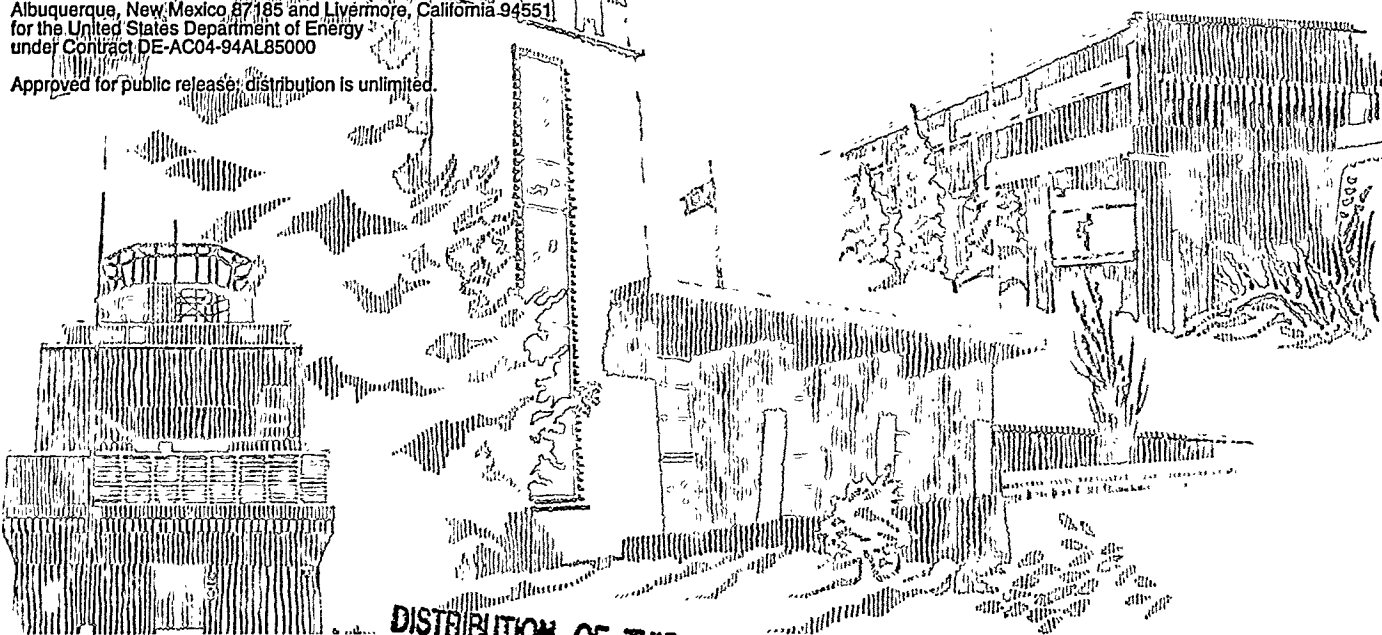
Phase I Prototype Final Report Part 2 - Domain Analysis

David Butler, David Eddy, Donna Edwards, Richard Judson, Robert Mariano,
William Mason, Leonard Napolitano, Leonard Schlessinger, Ann Yoshimura

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94551
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

MASTER



DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of the contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors or subcontractors.

This report has been reproduced from the best available copy.

Available to DOE and DOE contractors from:

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge TN 37831

Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from:

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.
Springfield, VA 22161

SAND97-8208
Unlimited Release
Printed November 1996

**KAISER PERMANENTE / SANDIA NATIONAL HEALTH CARE MODEL
Phase I Prototype Final Report
Part 2 - Domain Analysis**

Donna Edwards and Ann Yoshimura
Systems Research Department
Sandia National Laboratories / California
and
David Butler, Richard Judson, William Mason, and Leonard Napolitano
Scientific Computing Department
Sandia National Laboratories / California
and
Robert Mariano
Information Based Manufacturing Systems Department
Sandia National Laboratories / California
and
David Eddy and Leonard Schlessinger
Kaiser Permanente
Pasadena, California

ABSTRACT

This report describes the results of a Cooperative Research and Development Agreement (No. 01179W) between Sandia National Laboratories and Kaiser Permanente Southern California to develop a prototype computer model of Kaiser Permanente's health care delivery system. As a discrete event simulation, SimHCO models for each of 100,000 patients the progression of disease, individual resource usage, and patient choices in a competitive environment. SimHCO is implemented in the object-oriented programming language C++, stressing reusable knowledge and reusable software components. The versioned implementation of SimHCO showed that the object-oriented framework allows the program to grow in complexity in an incremental way. Furthermore, timing calculations showed that SimHCO runs in a reasonable time on typical workstations, and that a second phase model will scale proportionally and run within the system constraints of contemporary computer technology.

This report is published as two documents: Model Overview and Domain Analysis. A separate Kaiser-proprietary report contains the Disease and Health Care Organization Selection Models.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

INTRODUCTION

This report is Part 2 in a series of two reports describing the results of a Cooperative Research and Development Agreement between Sandia National Laboratories and Kaiser Permanente of Southern California to develop a prototype computer model of Kaiser Permanente's health care delivery system. The two documents are Model Overview and Domain Analysis. A separate Kaiser-proprietary report contains the Disease and Health Care Organization Selection Models.

DESCRIPTION OF SPECIFICATION FORMAT

This section describes the format of the class specifications that constitute the bulk of this document. The class specifications are organized into *clusters*; each cluster is a group of classes which are closely related to each other in some way. Within a cluster, the classes are categorized as *abstract*, *mutable*, or *immutable*. Abstract classes specify a class interface, that is the features and operations of a class, but do not specify any implementation of those features. A mutable representation specifies an implementation of an abstract class which has additional features that allow an instance of the class to be changed once it has been created. Immutable representations specify an implementation which can not be changed.

Whether abstract, mutable, or immutable, each class is specified by filling in the following specification template:

Class <NAME>

Description:

<description of purpose and defining characteristics of the class>

Specializes:

<parent>: <TYPE>

<description of the more general class and its role in the current class>

Aggregates:

<part>: <TYPE>

<description of the component class and its role in the current class>

Operations:

<operation>: <RESULT TYPE>

Purpose:

<description of the purpose of the operation>

Require:

<conditions which must be true before the operation can be invoked>

Algorithm:

<pseudocode for operation>

Ensure:

<conditions which are guaranteed to be true after the operation completes>

Invariants:

<invariant name>

<condition which is guaranteed to be true at all times after the creation of an instance of the class, except while an operation is actually executing>

Comments:

<comment topic>

<text of comment>

The *Specializes* section contains one *parent* clause for each class which the current class inherits. The *Aggregates* section contains one *part* clause for each part or component of the current class. Similarly, the *Operations*, *Invariants*, and *Comments* sections each contain as many *operation*, *invariant*, and *comment* clauses, respectively, as needed.

The *Invariant* section, and the *Require* and *Ensure* subclauses of the *operation* clause provide a precise method for specifying the intended operation and behavior of a class without actually implementing it. As described in the template above, these clauses provide conditions, written as precise logical expressions, which the class must satisfy. These conditions can be used both when reviewing the specification for correctness and when programming code to implement the specification.

Brief Table of Contents

GENERAL.....	15
SYSTEM CLUSTER.....	16
POPULATION CLUSTER.....	17
DISEASE CLUSTER.....	19
GROUP CLUSTER.....	19
HEALTH CARE ORGANIZATION CLUSTER.....	20
HEALTH PLAN CLUSTER.....	21
MEDICAL GROUP CLUSTER.....	24
HOSPITAL CLUSTER.....	36
FACILITY CLUSTER.....	38
DEPARTMENT CLUSTER.....	67
PERSONNEL CLUSTER.....	81
EQUIPMENT CLUSTER.....	100
SUPPLY CLUSTER.....	116
DRUG CLUSTER.....	131
UTILITY CLUSTER.....	137
RECORDS CLUSTER.....	143
ACCOUNTS CLUSTER.....	145

Table of Contents

GENERAL	15	HEALTH PLAN CLUSTER	21
GENERAL DESIGN ISSUES.....	15	ABSTRACT CLASSES.....	21
<i>Cost tracking and accounts</i>	15	<i>Class HEALTH_PLAN</i>	21
<i>Procedures and Interventions</i>	15	MUTABLE REPRESENTATIONS.....	24
SYSTEM CLUSTER	16	IMMUTABLE REPRESENTATIONS.....	24
ABSTRACT CLASSES.....	16	MEDICAL GROUP CLUSTER	24
<i>Class REGION</i>	16	ABSTRACT CLASSES.....	24
<i>Class HEALTH_CARE_SYSTEM</i>	16	<i>Class MEDICAL_GROUP</i>	24
MUTABLE REPRESENTATIONS.....	17	<i>Class PHYSICIAN</i>	26
IMMUTABLE REPRESENTATIONS.....	17	MUTABLE REPRESENTATIONS.....	28
POPULATION CLUSTER	17	<i>Class AIDS_SPECIALIST</i>	28
ABSTRACT CLASSES.....	17	<i>Class ANESTHIOLOGIST</i>	29
<i>Class POPULATION</i>	17	<i>Class CARDIOLOGIST</i>	29
<i>Class PERSON</i>	18	<i>Class CARDIOVASCULAR_SURGEON</i>	30
MUTABLE REPRESENTATIONS.....	19	<i>Class GYNECOLOGIST</i>	30
IMMUTABLE REPRESENTATIONS.....	19	<i>Class INTERNIST</i>	32
DISEASE CLUSTER	19	<i>Class NEONATOLOGIST</i>	32
ABSTRACT CLASSES.....	19	<i>Class OBSTETRICIAN</i>	32
MUTABLE REPRESENTATIONS.....	19	<i>Class PEDIATRICIAN</i>	33
IMMUTABLE REPRESENTATIONS.....	19	<i>Class PRIMARY_CARE_PHYSICIAN</i>	34
GROUP CLUSTER	19	<i>Class RADIOLOGIST</i>	35
ABSTRACT CLASSES.....	19	IMMUTABLE REPRESENTATIONS.....	35
<i>Class GROUP</i>	19	HOSPITAL CLUSTER	36
MUTABLE REPRESENTATIONS.....	20	ABSTRACT CLASSES.....	36
IMMUTABLE REPRESENTATIONS.....	20	<i>Class HOSPITAL</i>	36
HEALTH CARE ORGANIZATION CLUSTER	20	MUTABLE REPRESENTATIONS.....	38
ABSTRACT CLASSES.....	20	IMMUTABLE REPRESENTATIONS.....	38
<i>Class HEALTH_CARE_ORGANIZATION</i>	20	FACILITY CLUSTER	38
<i>Class NON_PROFIT_GROUP</i>	20	DESIGN ISSUES.....	38
<i>Class PROFIT_NETWORK_IPA</i>	21	<i>Intervention decomposition</i>	38
<i>Class PROFIT_STAFF_IPA</i>	21	<i>Intervention return types</i>	38
<i>Class NETWORK_FEE</i>	21	ABSTRACT CLASSES.....	38
		<i>Class FACILITY</i>	38

Class PATIENT_FACILITY.....	39
MUTABLE REPRESENTATIONS.....	40
Class BLOOD_DRAW_ROOM.....	40
Class CARDIAC_CARE_UNIT.....	40
Class CLASS_ROOM.....	41
Class CORONARY_CARE_UNIT.....	44
Class DELIVERY_ROOM.....	44
Class DISTRIBUTION_ROOM.....	45
Class EKG_ROOM.....	45
Class EMERGENCY_ROOM.....	46
Class EXAMINATION_ROOM.....	46
Class HIGH_LEVEL_PROCEDURE_ROOM.....	47
Class HOSPITAL_ROOM.....	49
Class INTENSIVE_CARE_UNIT.....	49
Class LABORATORY.....	50
Class LOBBY.....	52
Class LOW_LEVEL_PROCEDURE_ROOM.....	52
Class MEDICAL_WARD.....	56
Class NEONATAL_INTENSIVE_CARE_UNIT.....	57
Class NURSES_STATION.....	57
Class OPERATING_ROOM.....	58
Class PARKING_LOT.....	59
Class PEDIATRIC_WARD.....	59
Class PHYSICIANS_OFFICE.....	60
Class RADIOLOGY.....	63
Class STORAGE.....	64
Class TELEMETRY_UNIT.....	64
Class TREADMILL_ROOM.....	65
Class ULTRASOUND_ROOM.....	65
Class X_RAY_ROOM.....	66
Class WAITING_ROOM.....	66
IMMUTABLE REPRESENTATIONS.....	67
DEPARTMENT CLUSTER.....	67
ABSTRACT CLASSES.....	67
Class DEPARTMENT.....	67
Class APPOINTMENT_DEPARTMENT.....	68
Class MEDICAL_DEPARTMENT.....	69
Class CYD_DEPARTMENT.....	71

MUTABLE REPRESENTATIONS.....	71
Class CARDIAC_CARE_UNIT.....	71
Class CARDIOLOGY_DEPARTMENT.....	71
Class EMERGENCY_DEPARTMENT.....	72
Class HIV_DEPARTMENT.....	73
Class OBSTETRICS_AND_GYNECOLOGY_DEPARTMENT.....	73
Class PATIENT_EDUCATION_DEPARTMENT.....	75
Class PEDIATRICS_DEPARTMENT.....	76
Class PHARMACY_DEPARTMENT.....	76
Class PRIMARY_CARE_DEPARTMENT.....	77
Class RADIOLOGY_DEPARTMENT.....	80
Class SAME_DAY_APPOINTMENT_DEPARTMENT.....	81
Class TELEMETRY_UNIT.....	81
IMMUTABLE REPRESENTATIONS.....	81
PERSONNEL CLUSTER.....	81
ABSTRACT CLASSES.....	81
Class PERSONNEL.....	81
MUTABLE REPRESENTATIONS.....	83
Class ADMINSTRATOR.....	83
Class CARDIAC_CARE_UNIT_NURSE.....	83
Class CARDIAC_CARE_UNIT_STAFF.....	84
Class CARDIAC_NURSE.....	84
Class CLERK.....	85
Class CRITICAL_CARE_NURSE.....	85
Class DELIVERY_ROOM_PERSONNEL.....	85
Class DIETICIAN.....	86
Class HIGH_LEVEL_TECHNICIAN.....	86
Class INPATIENT_PERSONNEL.....	87
Class INSTRUCTOR.....	88
Class LAB_TECHNICIAN.....	90
Class LOGISTICAL_SUPPORT.....	92
Class LOW_LEVEL_TECHNICIAN.....	92
Class NEONATAL_INTENSIVE_CARE_UNIT_PERSONNEL.....	93
Class NON_MEDICAL_SUPPORT.....	93
Class NURSE.....	94
Class NURSE_PRACTITIONER.....	95
Class OPERATING_ROOM_PERSONNEL.....	96
Class PHARMACIST.....	96

Class PHLEBOTOMIST	97
Class RECEPTIONIST	97
Class SURGICAL_STAFF	98
Class TELEMETRY_UNIT_NURSE	99
Class TELEMETRY_UNIT_STAFF	99
IMMUTABLE REPRESENTATIONS	100
EQUIPMENT CLUSTER.....	100
ABSTRACT CLASSES	100
Class EQUIPMENT	100
Class DIAGNOSTIC_EQUIPMENT	101
MUTABLE REPRESENTATIONS	101
Class ABORTION_EQUIPMENT	101
Class AMNIOCENTESIS_EQUIPMENT	102
Class ANGIOGRAM_EQUIPMENT	102
Class ANGIOPLASTY_EQUIPMENT	103
Class BLOOD_DRAWING_KIT	103
Class	
BLOOD_OXYGEN_AND_NUTRIENT_MEASURING_EQUIPMENT	103
Class CARDIAC_BYPASS_EQUIPMENT	104
Class CARDIAC_CARE_UNIT_EQUIPMENT	104
Class CD4_TCELL_TEST_ANALYSIS_EQUIPMENT	105
Class CLASS_ROOM_EQUIPMENT	105
Class DELIVERY_ROOM_EQUIPMENT	106
Class ELISA_EQUIPMENT	107
Class EKG_MACHINE	107
Class HIV_ANTI_BODY_TEST_ANALYSIS_EQUIPMENT	108
Class INPATIENT_EQUIPMENT	108
Class INTENSIVE_CARE_UNIT_EQUIPMENT	109
Class MILER_PACKAGE	109
Class NEONATAL_INTENSIVE_CARE_UNIT_EQUIPMENT	109
Class PHYSICIANS_OFFICE_EQUIPMENT	110
Class OPERATING_ROOM_EQUIPMENT	111
Class PCR_ANALYSIS_EQUIPMENT	111
Class PCR_TEST_EQUIPMENT	111
Class SCALE	112
Class SHYGMOMANOMETER	112
Class TELEMETRY_UNIT_EQUIPMENT	113
Class THERMOMETER	113

Class THROMBOLYTIC_ADMINISTERING_EQUIPMENT	113
Class TREADMILL_EKG	114
Class ULTRASOUND_EQUIPMENT	114
Class URINE_SAMPLE_ANALYSIS_EQUIPMENT	115
Class WESTERN_BLOT_TEST_EQUIPMENT	115
Class X_RAY_EQUIPMENT	115
IMMUTABLE REPRESENTATIONS	116
SUPPLY CLUSTER.....	116
ABSTRACT CLASSES	116
Class SUPPLY	116
MUTABLE REPRESENTATIONS	117
Class ABORTION_SUPPLIES	117
Class AMNIOCENTESIS_SUPPLIES	117
Class ANGIOPLASTY_SUPPLIES	117
Class ANTI_TOBACCO_EDUCATION_HANDOUT	118
Class BLOOD_DRAWING_SUPPLIES	118
Class BLOOD_OXYGEN_AND_NUTRIENT_MEASURING_SUPPLIES	118
Class BLOOD_SAMPLE	119
Class CARDIAC_BYPASS_SURGERY_SUPPLIES	119
Class CD4_TCELL_TEST_ANALYSIS_SUPPLIES	120
Class CHOLESTEROL_TEST_SUPPLIES	121
Class CLEAN_NEEDLE_EDUCATION_HANDOUT	121
Class CONDOM	121
Class CORONARY_CARE_UNIT_SUPPLIES	122
Class DELIVERY_ROOM_SUPPLIES	122
Class DIET_EDUCATION_HANDOUT	122
Class ELISA_SUPPLIES	123
Class INPATIENT_SUPPLIES	123
Class INTENSIVE_CARE_UNIT_SUPPLIES	124
Class MATERNAL_ANTI_TOBACCO_EDUCATION_HANDOUT	124
Class MATERNAL_DRUG_EDUCATION_HANDOUT	125
Class MATERNAL_HIV_EDUCATION_HANDOUT	125
Class NEEDLE	125
Class NEONATAL_INTENSIVE_CARE_UNIT_SUPPLIES	126
Class OPERATING_ROOM_SUPPLIES	126
Class PCR_SUPPLIES	127
Class PRENATAL_EDUCATION_HANDOUT	127
Class SEX_EDUCATION_HANDOUT	127

Class SPONGE.....	128
Class TELEMETRY_UNIT_SUPPLIES.....	128
Class THROMBOLYTIC_SUPPLIES.....	128
Class ULTRASOUND_SUPPLIES.....	129
Class URINE_SAMPLE_SUPPLIES.....	129
Class WESTERN_BLOT_SUPPLIES.....	129
Class X-RAY_CONTRAST_AGENT_SUPPLIES.....	130
Class X-RAY_SUPPLIES.....	130
IMMUTABLE REPRESENTATIONS	131
DRUG CLUSTER	131
DESIGN ISSUES	131
<i>Intervention specific operations</i>	131
ABSTRACT CLASSES	131
Class DRUG.....	131
MUTABLE REPRESENTATIONS.....	131
Class AIDS_DRUG.....	131
Class ANTI_HYPERTENSIVE_DRUG.....	132
Class ANTIBIOTIC.....	132
Class ASPIRIN.....	133
Class AZT	133
Class BETA_BLOCKER.....	133
Class CALCIUM_CHANNEL_MEDICATION.....	134
Class CARDIAC_BYPASS_SURGERY_DRUG.....	134
Class CESAREAN_SECTION_DRUG.....	135
Class GANCOCLOVIR.....	135
Class LOVASTATIN	135
Class NIACIN.....	136
Class NITRATE.....	136
Class STREPTOKINASE.....	136
Class t-PA.....	137
IMMUTABLE REPRESENTATIONS	137
UTILITY CLUSTER	137
ABSTRACT CLASSES	137
Class NAME.....	137
Class ADDRESS (<ABBREVIATION>).....	138
Class MONEY.....	138
Class CALENDAR.....	138

Class SCHEDULE.....	138
Class TIME.....	140
Class DATE.....	140
Class QUEUE[].....	140
Class SET[]	142
Class BOUNDED_SET[]	143
MUTABLE REPRESENTATIONS	143
IMMUTABLE REPRESENTATIONS	143
RECORDS CLUSTER	143
ABSTRACT CLASSES	143
Class RECORD.....	143
Class EKG_RECORD.....	143
Class BLOOD_PRESSURE_RECORD.....	144
Class PATIENT_CHART	144
MUTABLE REPRESENTATIONS	144
IMMUTABLE REPRESENTATIONS	144
ACCOUNTS CLUSTER	145
DESIGN ISSUES	145
<i>Accounting model</i>	145
ABSTRACT CLASSES	145
Class BALANCE.....	145
Class ACCOUNTABLE.....	146
Class RESOURCE	147
Class SINGLE_USER_RESOURCE.....	149
Class REUSEABLE_RESOURCE.....	149
Class EXHAUSTABLE.....	150
Class CAPACITY	150
MUTABLE REPRESENTATIONS	151
Class ACCOUNTING_SYSTEM.....	151
Class SUMMARY_ACCOUNT.....	152
Class ACCOUNT.....	153
Class TRANSACTION.....	154
Class ACCOUNT_ENTRY	155
Class TRANSACTION1.....	155
Class TRANSACTIONIN.....	157
IMMUTABLE REPRESENTATIONS	158

GENERAL

The document constitutes a reorganization and translation of several Kaiser domain analysis documents [, ,] into an initial object-oriented analysis and design specification.

General design issues

Cost tracking and accounts.

Every component has cost tracking associated with it. Hence we need to design some general accounting method that is integrated into the components in some very uniform way. We've tentatively adopted the following mechanism. Costs are associated with resources. Each resource has four types of costs associated with it: acquisition cost, usage cost per unit of capacity, time cost per unit of time, and liquidation cost. Each resource has a usageCost function that calculates the cost for a usage instance. Each resource has one rate account and one cumulative account that track the costs associated with the resource.

Each resource has one or more procedures that it can perform and each procedure posts a transaction representing the value flows associated with the procedure. In general, resources are aggregate objects with other resources as parts and, in general, the procedures of the aggregate invoke the procedures of the parts. Each resource thus need account only for its own intrinsic costs; the parts will take care of their own costs. To account for its costs, the resource must post a transaction representing the flow of value from the resource to the user of the resource, which is typically the patient. For single user resources, such transactions are thus binary transactions, i.e. they have a single source, the resource, and a single destination,

the user of the resource. The standard process for accounting is thus that when usage of a resource begins, the time is recorded and when usage ends, the usageCost function is invoked and the usage transaction posted. This pattern is part of every procedure, even if not explicitly stated in the specification of the procedure. For more details see class *RESOURCE* and the discussion and specifications in the *ACCOUNT* cluster.

Procedures and Interventions

Procedures and interventions are central activities in the function of the health care organization, but they are not objects. They are actions taken by the various objects in the model and thus they appear as operations of the relevant objects. For instance, the anti-tobacco education intervention is something that a physician (among other objects) does, hence the *PHYSICIAN* class has a *giveAntiTobaccoEducation* operation. In general, procedures and interventions take place in some facility and require coordinated behavior among various resources - patients, physicians, other personnel, equipment, supplies, drugs, etc. A given intervention may thus appear as an operation on several of the resources involved. In the case of anti-tobacco education, the *giveAntiTobaccoEducation* operation appears in both the *PHYSICIANS_OFFICE* class and the *PHYSICIAN* class. *PHYSICIANS_OFFICE.giveAntiTobaccoEducation* ensures that various resources (e.g. the doctor, the patient, handouts, etc) are available, then passes them to *PHYSICIAN.giveAntiTobaccoEducation*, which performs the intervention. This is the typical structure. For more details, see the specification for *PHYSICIAN* and *PHYSICIANS_OFFICE*

SYSTEM CLUSTER

Abstract classes

Class *REGION*

Description:

The geographical region of interest, including its physical, environmental, economic, and demographic features.

Specializes:

None

Aggregates:

boundary: GEOGRAPHIC_BOUNDARY

The boundary of the region

people: POPULATION

The human population of the region

healthCare: HEALTH_CARE_SYSTEM

The health care system of the region

environmentalHealthFactors: <TYPE>

Some collection of regional features which influence the health of the population of the region. To be specified.

financialFactors: <TYPE>

Some collection of regional features which influence the finances of the region. To be specified.

jobFactors: <TYPE>

Some collection of regional features which influence jobs in the region. To be specified.

Operations:

Invariants:

Comments:

Environment

The *environmentalHealthFactors*, *financialFactors*, and *jobFactors* features are just place holders for whatever specific features are eventually identified for class *REGION*. This feature corresponds to the category "External Environment" in KDA5.2. See also feature *demographicFactor* in class *POPULATION*

Class *HEALTH_CARE_SYSTEM*

Description:

The general, abstract health care system

Specializes:

None

Aggregates:

people: POPULATION

The population the health care system serves, presumably the population of the region. See comment below.

groups: SET[EMPLOYER_GROUP]

The collection of employer groups in the health care system. See comment below.

hcos: SET[HEALTH_CARE_ORGANIZATION]

The collection of HCOs operating in the region.

regulators: SET[REGULATOR]

the collection of regulators in the region.

suppliers: SET[SUPPLIER]

The collection suppliers to the health care system. Includes banks, which supply financing.

Operations:

Invariants:

Comments:

Other Groups

KDA5.2 defines general groups as, roughly speaking, arbitrary sets of people. General groups then seem to be separated into two categories: employer groups and groups

POPULATION CLUSTER

Abstract classes

Class *POPULATION*

Description:

A human population.

Specializes:

SET[PERSON]

The general, abstract set of PERSON objects.

of people which meet some specification (e.g. age < 30). An employer group is certainly an object with identity and behavior and hence I've specified an EMPLOYER_GROUP class. It's not clear to me that entities of the other category have identity, existence and behavior. Their more like results of a query - just a container full of references to people that match the query specification.

Population

The population in the region and the population in the health care system are distinct. Do we want to bind them together with an invariant?

Mutable representations

Immutable representations

Aggregates:

demographicFactor: DISTRIBUTION_FUNCTION

Some statistical feature of the population. See comment below.

Operations:
Invariants:
Comments:

Features and distributions

Whatever features we define for PERSON (see below) conceptually propagate to the population as distribution functions. In addition, KDA5.2 states that there are "environmental health factors" and "demographic factors", but does not specify an explicit set. These unspecified features presumably appear as externally imposed distributions on the population. We need to detail these features.

Class PERSON

Description:

The abstract, general human being

Specializes:

ACCOUNTABLE

The general object which can store value.

Aggregates:

name: NAME

address: ADDRESS

hcoMembership: HCO

The health care organization the person belongs to.

employerGroup: GROUP

The employer group the person belongs to.

coverageType: COVERAGE_POLICY

The type of coverage policy the person has.

healthCare: ACCOUNT

The health care value received by the patient.

income: MONEY

The annual income of the person.

language: STRING

The language the person speaks

age: INTEGER

The age of the person (in sec?)

gender: ENUM {MALE, FEMALE}

the gender of the person

familyHistory: <TYPE>

The family history of the person.

cvdPredisposition: REAL

The predisposition to coronary artery occlusion.

Operations:

getExamination: EXAM_RECORD

Purpose:
Receive medical examination

Require:
None

Algorithm:
deferred

Ensure:
None

Invariants:

Comments:

identifiers

We can have some structure like SET[ID] which must be bound to data at execution time, or we can define specific

DISEASE CLUSTER

Abstract classes

Mutable representations

Immutable representations

GROUP CLUSTER

Abstract classes

Class ***GROUP***

Description:

The general, abstract employer group.

identifiers. The latter is gerneally the better policy, but requires some specific choice. I'll assume the latter.

Values of characteristics

I interpret the value types given in the KDA as follows: "dichotomous" and "categorical" either as specific types (RACE) or as ENUMERATED; "counts" as INTEGER; "continuous" as REAL.

Static versus variable characteristics

All features are variable in principle. If no operation specifically changes them, they are static.

Family history

How do we represent family history? Doesn't genetic predisposition subsume family history?

Mutable representations

Immutable representations

Specializes:
Aggregates:
Operations:
Invariants:
Comments:

HEALTH CARE ORGANIZATION CLUSTER

Abstract classes

Class *HEALTH_CARE_ORGANIZATION*

Description:

<description of purpose and defining characteristics of the class>

Specializes:
Aggregates:

healthPlan: HEALTH_PLAN

the health plan part of the HCO

medicalGroup: MEDICAL_GROUP

The medical group part of the HCO

hospital: HOSPITAL

The hospital part of the HCO

Differentiating features

What specific differentiating features does this class have?

Mutable representations
Immutable representations

Operations:
Invariants:
Comments:

Number of health plans, medical groups, and hospitals.

If taken literally, KDA5.2 specifies that each HCO has a single health plan, medical group, and hospital. It seems unlikely this is correct. What's the right specification?

Class *NON_PROFIT_GROUP*

Description:

A non-profit group health care organization.

Specializes:

HEALTH_CARE_ORGANIZATION

The general health care organization

Aggregates:
Operations:
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PROFIT_NETWORK_IPA*

Description:

A non-profit group health care organization.

Specializes:

HEALTH_CARE_ORGANIZATION

The general health care organization

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PROFIT_STAFF_IPA*

Description:

A non-profit group health care organization.

Specializes:

HEALTH_CARE_ORGANIZATION

The general health care organization

HEALTH PLAN CLUSTER

Abstract classes

Class *HEALTH_PLAN*

Description:

The general, abstract health plan part of an HCO

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *NETWORK_FEE*

Description:

A non-profit group health care organization.

Specializes:

HEALTH_CARE_ORGANIZATION

The general health care organization

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Mutable representations

Immutable representations

Specializes:

ACCOUNTABLE

The general object which can store value

Aggregates:

groups: SET[GROUP]

The customers of the plan.

memberCt: INTEGER

The number of members

suppliers: SET[SUPPLIER]

The collection of suppliers to this health plan.

budget: MONEY

The annual operating budget for the health plan. See comment below.

accountBalance: MONEY

The current account balance. See comment below.

ytdRevenues: MONEY

The year to date revenues of the health plan. See comment below.

ytdExpenses: MONEY

The year to date expenses of the health plan. See comment below.

rateSetting: PROGRAM

The rate setting program (i.e. temporary working group) for the health plan.

coveragePolicies: SET[COVERAGE_POLICY]

The collection of coverage policies offered by the plan.
Operations:

setRates

Purpose:
sets the rates for all coverage policies

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

setCoveragePolicies

Purpose:
defines all coverage policies

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

setFinancialPolicies

Purpose:
defines all financial policies

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

collectRevenues

Purpose:
collects revenues from *groups*

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

paySuppliers

Purpose:
Pays the suppliers of the health plan.

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

payMedicalGroup

Purpose:
Pays the *medicalGroup* part of the HCO.

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

payHospital

Purpose:
Pays the *Hospital* part of the HCO.

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

Invariants:
Comments:

Revenues, expenses, balances and accounts

I've specified a single layer of aggregate financial features.
In principle, there should be some chart of accounts and all the usual accounting machinery. How little can we get away with?

MEDICAL GROUP CLUSTER

Abstract classes

Class *MEDICAL_GROUP*

Description:

The medical group component of a health care organization; provides physicians to the HCO

Specializes:

ACCOUNTABLE

The general object which can store value.

Aggregates:

staff: SET[PHYSICIAN]

The medical staff of the medical group.

productivity: REAL

The (average?) number of patients seen per outpatient hour.

Financial policies

I've not specified any feature for financial policies because it's not clear how to deal with them.

Groups

Shouldn't HEALTH_PLAN have a feature groups:SET[GROUP]? Aren't rates specific to group?
Mutable representations
Immutable representations

revenue: MONEY

The revenue received from the health plan.

expense: MONEY

The expenses paid to physicians.

profit: MONEY

revenue - expense

hoursOfOperation: SCHEDULE

<description of the component class and its role in the current class>

physicianCompensation: <TYPE?>

The physician compensation policy. See comment below

practiceGuidelineDevelopment: PROGRAM

The program for developing practice guidelines

practiceGuidelineImplementation: PROGRAM

The program for implementing practice guidelines
Operations:

setHours: VOID

Purpose:
Sets the policy for hours of operation
Require:
<preconditions of the operation>
Algorithm:
deferred
Ensure:
<postconditions for the operation>

setVisitSchedules: VOID

Purpose:
Sets the schedules for outpatient visits, which
presumably influence *productivity*
Require:
<preconditions of the operation>
Algorithm:
deferred

Ensure:
<postconditions for the operation>

setCompensation: VOID

Purpose:
Sets the salaries, bonuses, and retirement benefits for
physicians.

Require:
<preconditions of the operation>

Algorithm:
<pseudocode for operation>

Ensure:
<postconditions for the operation>

setPracticeGuidelines: VOID

Purpose:
Sets the practice guidelines for various interventions.

Require:
<preconditions of the operation>

Algorithm:
<pseudocode for operation>

Ensure:
<postconditions for the operation>
Invariants:

Profit

profit = revenue - expense

Comments:

Physician compensation

How do we want to specify physician compensation (salary, bonus, retirement)? Is it specified as a range based on specialty and experience? How do we choose the values for each physician?

Practice guidelines

What are the specific practice guidelines we need to implement? How do we plan to represent them?

Practice guideline programs

How are these used? Is there just one of each, or are there several active concurrently? (e.g. one for each guideline under development?)

Class *PHYSICIAN*

Description:

The general, abstract physician
Specializes:

PERSONNEL

The general, abstract personnel.

Aggregates:
Operations:

examinePatient(p: PERSON, c: CHART): VOID

Purpose:
examine patient p

Require:
p not void; chart not void

Algorithm:
examRecord = patient.getExamination
chart.insert(examRecord)

Ensure:
chart.contains(examRecord)

giveAntiTobaccoEducation(patient: PERSON, handout: ANTI_TOBACCO_EDUCATION_HANDOUT, equipment: SET[EQUIPMENT]): VOID

Purpose:
Give anti-tobacco education to patient, using handout and equipment.

Require:
patient != void
handout != void
equipment != void

Algorithm:
patient.receiveAntiTobaccoEducation(handout)

Ensure:
patient receives education

**giveDietEducation(patient: PERSON,
handout:DIET_EDUCATION_HANDOUT
T, equipment: SET[EQUIPMENT]): VOID**

Purpose:
Give diet education to patient using handout and
equipment

Require:
patient != void
handout != void
equipment != void

Algorithm:
deferred

Ensure:
patient receives education

**giveCleanNeedleEducation(patient: PERSON,
handout:CLEAN_NEEDLE_EDUCATIO
N_HANDOUT,
SET[EQUIPMENT]): VOID**

Purpose:
Give clean needle education to patient using handout
and equipment

Require:
patient != void
handout != void
equipment != void

Algorithm:
deferred

Ensure:
patient receives education

**giveSexEducation(patient: PERSON,
handout:SEX_EDUCATION_HANDOUT,
equipment: SET[EQUIPMENT]): VOID**

Purpose:
Give sex education to patient using handout and
equipment

Require:
patient != void
handout != void
equipment != void

Algorithm:
deferred

Ensure:
patient receives education

**givePrenatalEducation(patient: PERSON,
handout:PRENATAL_EDUCATION_HA
NDOUT,
SET[EQUIPMENT]): VOID**

Purpose:
Give prenatal education to patient using handout and
equipment

Require:
 patient /= void
 handout /= void
 equipment /= void

Algorithm:
 deferred

Ensure:
 patient receives education

giveMaternalAntiTobaccoEducation: <RESULT TYPE>

 patient chart

 All these operations need to be re-examined for updating patient chart.

 Mutable representations
 Class *AIDS_SPECIALIST*
 Description:

 An AIDS specialist.
 Specializes:

PHYSICIAN

 The general physician
 Aggregates:
 Operations:

performLymphNodeSizeTest: <RESULT TYPE>

 Purpose:
 To be specified

treatCMV: <RESULT TYPE>

 Purpose:
 To be specified

 generalized education

 An alternative to all these specialized education operations is the specify a general education class and several specializations, antiTobaccoEducation etc. then the

treatPCP: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *ANESTHIOLOGIST*

Description:

A specialist in anesthesiology
Specializes:

PHYSICIAN

The general physician

Aggregates:
Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CARDIOLOGIST*

Description:
A cardiology specialist.
Specializes:

PHYSICIAN

The general physician

Aggregates:
Operations:

performAngiogram: <RESULT TYPE>

Purpose:
To be specified

performAngioplasty: <RESULT TYPE>

Purpose:
To be specified

performTreadmillTest: <RESULT TYPE>

Purpose:
To be specified

treatMIWithCCU: <RESULT TYPE>

Purpose:
To be specified

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:
To be specified

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:
To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:
To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:
To be specified

treatWithNiacin: <RESULT TYPE>

Purpose:
To be specified

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **CARDIOVASCULAR_SURGEON**

Description:

A specialist in cardiovascular surgery.

Specializes:

PHYSICIAN

The general physician

Aggregates:

Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:
To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **GYNECOLOGIST**

Description:

A gynecology specialist.

Specializes:
Aggregates:
Operations:

measure21TrisomyWithAmniocentesis: <RESULT
TYPE>

Purpose:
To be specified

measureBloodOxygenAndNutrientsInPlacenta:
<RESULT TYPE>

Purpose:
To be specified

measureFetusSizeWithUltrasound: <RESULT
TYPE>

Purpose:
To be specified

treat21TrisomyFetus: <RESULT TYPE>

Purpose:
To be specified

treatCesarean: <RESULT TYPE>

Purpose:
To be specified

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:
To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

treatMaternalHIV: <RESULT TYPE>

Purpose:
To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *INTERNIST*

Description:

An internal medicine specialist.

Specializes:

PHYSICIAN

The general physician

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *NEONATOLOGIST*

Description:

A specialist in neonatology.

Specializes:

PHYSICIAN

The general physician

Aggregates:
Operations:
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *OBSTETRICIAN*

Description:

An obstetrics specialist.

Specializes:

PHYSICIAN

The general physician

Aggregates:

Operations:

measure21TrisomyWithAmniocentesis: *<RESULT
TYPE>*

Purpose:

To be specified

measureBloodOxygenAndNutrientsInPlacenta:
<RESULT TYPE>

Purpose:

To be specified

measureFetusSizeWithUltrasound: <RESULT TYPE>

Purpose:
To be specified

treat21TrisomyFetus: <RESULT TYPE>

Purpose:
To be specified

treatCesarean: <RESULT TYPE>

Purpose:
To be specified

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:
To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

treatMaternalHIV: <RESULT TYPE>

Purpose:
To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PEDIATRICIAN*

Description:

A specialist in pediatrics.

Specializes:

PHYSICIAN

The general physician

Aggregates:
Operations:

treatCesarean: <RESULT TYPE>

Purpose:
To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PRIMARY_CARE_PHYSICIAN*

Description:

A primary care physician.

Specializes:

PHYSICIAN

The general physician

Aggregates:
Operations:

treatWithAspirin: <RESULT TYPE>

Purpose:
To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:
To be specified

treatWithNiacin: <RESULT TYPE>

Purpose:
To be specified

treatCMV: <RESULT TYPE>

Purpose:
To be specified

treatMaternalHIV: <RESULT TYPE>

Purpose:
To be specified

treatPCP: <RESULT TYPE>

Purpose:
To be specified

performLymphNodeSizeTest: <RESULT TYPE>

Purpose:
To be specified

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **RADIOLOGIST**

Description:

A radiology specialist.

Specializes:

PHYSICIAN

The general physician

Aggregates:
Operations:

performAngioplasty: <RESULT TYPE>

Purpose:
To be specified

measurePelvisSizeWithXRay: <RESULT TYPE>

Purpose:
To be specified

measurePelvisSizeWithXRay: <RESULT TYPE>

Purpose:
To be specified

performAngiogram: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?
Immutable representations

HOSPITAL CLUSTER

Abstract classes

Class *HOSPITAL*

Description:

The general abstract hospital

Specializes:

None

Aggregates:

inPatientService: TYPE?

The in-patient service subsystem. See comment below.

outPatientService: SET[DEPARTMENT]

The out-patient service subsystem.

inPatientDays: REAL

The number of in-patient days per thousand members

normalBirthStay: INTEGER

The average length of stay for a normal birth.

phoneHoldingTime: TIME

The average holding time for telephone access.

apptHoldingTime: TIME

The average holding time for a routine appointment.

parkingAvailability: REAL

Some measure of the average availability of parking.

facilities: SET[FACILITY]

The collection of facilities in the hospital.

personnel: SET[PERSONNEL]

The collection of non-physician personnel of the hospital.

equipment: SET[EQUIPMENT]

The collection of equipment used by the hospital.

supplies: SET[SUPPLY]

The collection of supplies used by the hospital.

drugs: SET[DRUG]

The collection of drugs used by the hospital.

Operations:

setFacilitySchedules: VOID

Purpose:
Set schedules (policy) for facilities.

Require:
<preconditions of the operation>

Algorithm:
<pseudocode for operation>

Ensure:
For all f such that facilities.contains(f), not
f.schedule.void

receivePayment(p:MONEY): VOID

Purpose:
Accept payment from HEALTH_PLAN for resources
provided. See comment on accounting below.

Require:
not p.void

Algorithm:
deferred

Ensure:
<postconditions for the operation>

paySuppliers: VOID

Purpose:
Make payment to suppliers. See comment on
accounting below.

Require:
<preconditions of the operation>

Algorithm:
deferred

Ensure:
<postconditions for the operation>

Invariants:

Comments:

InPatientService

The structure of the in-patient service subsystem has yet to
be specified.

**normalBirthStay, apptHoldingTime,
apptHoldingTime**

Are these some sort of average (as I've specified above) or
are they a predetermined, policy driven value?

Collections versus specific parts.

The various parts of the hospital (facilities, personnel,
equipment, etc) have all been specified as abstract SETs,
i.e. arbitrary collections. Is this the way we want to model

it, or are these just place holders for some specific facilities, equipment, etc which we have yet to specify?

Accounting mechanism

Once we've designed some sort of accounting mechanism, we need to revisit receivePayment and paySuppliers to include the mechanism in the specification.

FACILITY CLUSTER

Design Issues

Intervention decomposition

In this design, interventions are implemented in layers. Each intervention operation in the facility layer is responsible for making sure all the necessary resources, e.g. patient, personnel, equipment and supplies, are present. It then passes these resources to the corresponding operation in the personnel layer, which uses the corresponding operations in the patient, equipment and supplies layer to perform the intervention. This architecture is reasonably faithful to the problem domain and also minimizes the coupling between the various layers, making the classes more reusable.

Intervention return types

We need a uniform policy for how the results of interventions, especially diagnostic interventions will be communicated. The policy specified here distinguishes between information results, e.g. blood pressure, and physical results, e.g. blood sample. Information results are uniformly stored in the patient's chart by the personnel layer intervention. Physical results are features (parts) of the facility which are passed to the personnel layer intervention and set by it. The facility layer intervention leaves the results in the facility, to be transferred by some other procedure. The presumption is that this is approximately how it really works in the problem domain.

Interaction with membership

We have yet to specify any interaction with the membership of the HCO. How do members use the HOSPITAL?

Mutable representations

Immutable representations

Abstract classes

Class *FACILITY*

Description:

The general, abstract facility: a building, room or other physical portion of a HOSPITAL.

Specializes:

REUSEABLE_RESOURCE

The abstract reusable resource.

Aggregates:

size: REAL

The size of the facility in square feet.

maintenancePersonnel: SET[PERSONNEL]

The personnel required to maintain the facility.

maintenanceEquipment: SET[EQUIPMENT]

The equipment required to maintain the facility.

maintenanceSupplies: SET[SUPPLIES]

The supplies required to maintain the facility.
Operations:

<opr>: <RESULT TYPE>

Purpose:
<description of the purpose of the operation>
Require:
<preconditions of the operation>
Algorithm:
<pseudocode for operation>
Ensure:
<postconditions for the operation>
Invariants:

<invariant>
<description of class invariant>
Comments:

Maintenance requirements and specialization

Once again I've used collections for the maintenance personnel, equipment and supplies. But aren't these just place holders? Doesn't each facility require specific maintenance roles filled by specific specialized types? For instance, an MRI suite must require specific maintenance personnel and supplies. If not, aren't these features common to all reusable resources and therefore can't they be moved to class REUSABLE_RESOURCE?

Class PATIENT_FACILITY

Description:
A facility that can service a single patient
Specializes:

FACILITY

The general facility

SINGLE_USER_RESOURCE

The general single user resource
Aggregates:

patient: PERSON

The patient currently in the room.

chart: PATIENT_CHART

The chart associated with *patient*.
Operations:

receivePatient(p: PERSON): VOID

Purpose:
Makes *p* the current patient.
Require:
p not void
Algorithm:
patient := p;

transaction.setDestination(patient.healthCare);
currentUsageStart := current time

Ensure:

See algorithm

releasePatient: PERSON

Purpose:

removes the patient from the facility

Require:

patient not void

Algorithm:

result := patient
patient := void
postUsageCost

Ensure:

See algorithm

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Mutable representations

Class *BLOOD_DRAW_ROOM*

Description:

A facility for drawing blood

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:

Operations:

performHivAntiBodyTest: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CARDIAC_CARE_UNIT*

Description:

A cardiac care unit

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:
Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:
To be specified

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CLASS_ROOM*
Description:

The general patient education class room.

Specializes:

FACILITY

The general facility.

Aggregates:

instructors: SET[TYPE?]

The current instructors. See comment below.

students: SET[PERSON]

The class; the current students in the class room.

antiTobaccoHandouts:

*SET[ANTI_TOBACCO_EDUCATION_H
ANDOUT]*

Handouts for the anti-tobacco classes.

dietHandouts:

SET[DIET_EDUCATION_HANDOUT]

Handouts for the diet classes.

cleanNeedleHandouts:

*SET[CLEAN_NEEDLE_EDUCATION_H
ANDOUT]*

Handouts for the clean needle education.

sexHandouts: SET[SEX_EDUCATION_HANDOUT]

Handouts for the sex education.

prenatalHandouts:

*SET[PRENATAL_EDUCATION_HAN
DOUT]*

Handouts for the prenatal education.

equipment: SETEQUIPMENT]

The standard class room equipment; see comment.

Operations:**giveAntiTobaccoEducation: VOID****Purpose:**

Give anti-tobacco education to *students*.

Require:

capacity ≥ 20
instructors.ct = 2
antiTobaccoHandouts.ct $\geq$ *students.ct*
equipment $\neq$ void

Algorithm:
deferred**Ensure:**

students received education

giveDietEducation: VOID**Purpose:**

Give diet education to *students*

Require:

capacity ≥ 20
instructors.ct = 1
instructors.item(1).conforms_to(DIETICIAN)
dietHandouts.ct $\geq$ *students.ct*
equipment $\neq$ void

Algorithm:

*instructors.item(1).giveDietEducation(students,
 dietHandouts, equipment)*

Ensure:

students receive diet education

giveCleanNeedleEducation: VOID**Purpose:**

Give clean needle education to *students*.

Require:

capacity ≥ 20
instructors.ct ≥ 1
cleanNeedleHandouts.ct $\geq$ *students.ct*
equipment $\neq$ void

Algorithm:

*instructors.item(1).giveCleanNeedleEducation(students,
 cleanNeedleHandouts, equipment)*

Ensure:

students received education

giveSexEducation: VOID**Purpose:**

Give sex education to *students*.

Require:

capacity ≥ 20
instructors.ct ≥ 1
sexHandouts.ct $\geq$ *students.ct*
equipment $\neq$ void

Algorithm:

*instructors.item(1).giveSexEducation(students,
sexHandouts, equipment)*

Ensure:

students received education

givePrenatalEducation: VOID

Purpose:

Give prenatal education to students.

Require:

*capacity >= 20
instructors.ct >= 1
prenatalHandouts.ct >= students.ct
equipment != void*

Algorithm:

*instructors.item(1).givePrenatalEducation(students,
prenatalHandouts, equipment)*

Ensure:

students received education

*giveMaternalAntiTobaccoEducation:
TYPE*

<RESULT

Purpose:

To be specified

giveMaternalDrugEducation: <RESULT TYPE>

Purpose:

To be specified

giveMaternalHivEducation: <RESULT TYPE>

Purpose:

To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:

To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Type for instructors

Instructors are required to have a number of operations of the form give???Education. I've put these operations in *PERSONNEL*, but it seems unlikely this is correct. What class do these operations go in?

Equipment

What specific equipment should be in every classroom?

giveAntiTobaccoEducation requirements

Can we be more specific about what equipment is required?

Handouts

This specification lists each different type of handout as a feature, as if the classroom had a rack or file with slots for each kind of handout. Alternately, there could be a single slot for generic handouts. The preconditions for the various operations would have to require that the proper handouts were present.

Class CORONARY_CARE_UNIT

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

Presumably the differenting features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class DELIVERY_ROOM

Description:

<description of purpose and defining characteristics of the class>

Specializes:

PATIENT_FACILITY

The general patient facility.

Aggregates:

Operations:

treatEclampsia: <RESULT TYPE>

Purpose:

To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:

To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

Presumably the differenting features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *DISTRIBUTION_ROOM*

Description:

A room for distributing needles and condoms.

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:

Operations:

distribution_room::distributeCleanNeedles:

<RESULT TYPE>

Purpose:

To be specified

distribution_room::distributeCondoms:

TYPE>

<RESULT

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *EKG_ROOM*

Description:

A room specifically for taking EKG measurements

Specializes:

FACILITY

The general facility

Aggregates:

patient: PERSON

The current patient in the room.

chart: PATIENT_CHART

The chart associated with *patient*.

personnel: PERSONNEL

The current personnel in the room

ekgMachine: EKG_MACHINE

The EKG machine currently in the room.

ekgTransaction: TRANSACTION11

The transaction corresponding to the performEKG procedure

Operations:

performEKG: VOID

Purpose:

perform an EKG measurement on *patient*

Require:

patient not void

personnel not void

personnel.conformsTo(PHLEBOTOMIST) or

personnel.conformsTo(LOW_LEVEL_TECHNICIAN)

CIAN

ekgMachine not void

Algorithm:

personnel.performEKG(patient, ekgMachine, chart)

Ensure:

chart.contains(ekg)

Invariants:

transaction destination

ekgTransaction.destination = patient.healthCare or (*patient*

void and *ekgTransaction.destination* void)

Comments:

Class EMERGENCY_ROOM

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

Operations:

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

Presumably the differing features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class EXAMINATION_ROOM

Description:

The general examination room

Specializes:

FACILITY

The general facility

Aggregates:

patient: PERSON

The patient currently in the room.

chart: PATIENT_CHART

The chart associated with *patient*.

doctor: PHYSICIAN

The doctor doing the examination.

Operations:

examinePatient: VOID

Purpose:

doctor examines *patient*

Require:

doctor not void and *patient* not void

Algorithm:

doctor.examinePatient(patient, chart)

Ensure:

chart contains examination record

measure21TrisomyWithAmniocentesis: **<RESULT
TYPE>**

Purpose:

To be specified

The general facility.

performLymphNodeSizeTest: **<RESULT TYPE>**

Purpose:

To be specified

measureFetusSizeWithUltrasound: **<RESULT
TYPE>**

Purpose:

To be specified

treatMaternalHIV: **<RESULT TYPE>**

Purpose:

To be specified

Invariants:

Comments:

Other procedures?

What's done in an examination room?

Class **HIGH_LEVEL_PROCEDURE_ROOM**

Description:

<description of purpose and defining characteristics of the
class>

Specializes:

FACILITY

Aggregates:

<role>; <TYPE>

<description of the component class and its role in the current class>

Operations:

measureBloodOxygenAndNutrientsInPlacenta:
 <RESULT TYPE>

Purpose:
 To be specified

measureFetusSizeWithUltrasound:
 <RESULT TYPE>

Purpose:
 To be specified

performAngiogram: <RESULT TYPE>

Purpose:
 To be specified

treat21TrisomyFetus: <RESULT TYPE>

Purpose:
 To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:
 To be specified

measure21TrisomyWithAmniocentesis:
 <RESULT TYPE>

Purpose:
 To be specified

performAngioplasty: <RESULT TYPE>

Purpose:
 To be specified

performTreadmillTest: <RESULT TYPE>

Purpose:
 To be specified
Invariants:

<invariant>

<description of class invariant>

Comments:

Differentiating features

Presumably the differing features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *HOSPITAL_ROOM*

Description:

An hospital inpatient room.

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:

Operations:

treat21TrisomyFetus: <RESULT TYPE>

Purpose:

To be specified

treatCesarean: <RESULT TYPE>

Purpose:

To be specified

treatCMV: <RESULT TYPE>

Purpose:

To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:

To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:

To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:

To be specified

treatPCP: <RESULT TYPE>

Purpose:

To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *INTENSIVE_CARE_UNIT*

Description:

An intensive care unit

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:

Operations:

treatEclampsia: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class LABORATORY

Description:

A laboratory for performing medical tests and analyses.

Specializes:

FACILITY

The general facility.

Aggregates:

personnel: SET[LAB_TECHNICIAN]

The staff of the lab.

cholesterolTestSupplies:

SET[CHOLESTEROL_TEST_SUPPLIES]

Supplies used to test blood for cholesterol.

Operations:

measureLDLCholesterol(sample:

BLOOD_SAMPLE): VOID

Purpose:

Analyse the LDL cholesterol level in *sample*.

Require:

personnel.ct >= 1

cholesterolTestSupplies.ct > 0

sample not void

Algorithm:

deferred

Ensure:

results returned to *sample.origin*

measureHDLCholesterol(sample:

BLOOD_SAMPLE): VOID

Purpose:

Analyse the HDL cholesterol level in *sample*.

Require:

personnel.ct >= 1

cholesterolTestSupplies.ct > 0

sample not void

Algorithm:
deferred

Ensure:
results returned to *sample.origin*

measureTotalCholesterol(sample;
BLOOD_SAMPLE); VOID

Purpose:
Analyse the total cholesterol level in *sample*.

Require:

personnel.ct >= 1
cholesterolTestSupplies.ct > 0
sample not void

Algorithm:
deferred

Ensure:
results returned to *sample.origin*

measure21TrisomyWithAmniocentesis; <RESULT
TYPE>

Purpose:
To be specified

measureBloodOxygenAndNutrientsInPlacenta;
<RESULT TYPE>

Purpose:
To be specified

performCd4TCellTest; <RESULT TYPE>

Purpose:
To be specified

performCMVTest; <RESULT TYPE>

Purpose:
To be specified

performFungusTest; <RESULT TYPE>

Purpose:
To be specified

performFungusTest; <RESULT TYPE>

Purpose:
To be specified

performHivAntiBodyTest; <RESULT TYPE>

Purpose:
To be specified

performHivPerTest; <RESULT TYPE>

Purpose:
To be specified

performLungOxygenTest: <RESULT TYPE>

Purpose:
To be specified

performPCPTTest: <RESULT TYPE>

Purpose:
To be specified

measureProteinLevelInUrine: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

Presumably the differing features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class LOBBY

Description:

The general lobby; a waiting room without many seats?

Specializes:

WAITING_ROOM

The general waiting room.

Aggregates:

FACILITY

The general facility.

Operations:

<opr>: <RESULT TYPE>

Purpose:
<description of the purpose of the operation>

Require:
<preconditions of the operation>

Algorithm:
<pseudocode for operation>

Ensure:
<postconditions for the operation>

Invariants:

<invariant>

<description of class invariant>

Comments:

Differentiating features

Is this class any different than waiting room?

Class LOW_LEVEL_PROCEDURE_ROOM

Description:

A room for performing low level procedures

Specializes:

FACILITY

The general facility.

Aggregates:

patient: PERSON

The current patient in the room.

chart: PATIENT_CHART

The chart associated with *patient*.

personnel: PERSONNEL

The current personnel in the room

bloodPressureGauge: SPHYGMOMANOMETER

equipment for measuring patient's blood pressure

bloodDrawingKit: BLOOD_DRAWING_KIT

A collection of equipment and/or supplies for drawing blood.

bloodDrawingSupplies:

SET[BLOOD_DRAWING_SUPPLIES]

Supplies for drawing blood.

bloodSample: BLOOD_SAMPLE

The blood sample drawn from the patient by the last drawBlood operation.

ekgMachine: EKG_MACHINE

The EKG machine currently in the room.

Operations:

processStandardMeasures: <RESULT TYPE>

Purpose:

<description of the purpose of the operation>

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

takeMedicalHistory: VOID

Purpose:

Take the medical history of *patient* using *chart*.

Require:

patient not void

personnel.conforms_to(NON_CRITICAL_CARE_NURSE)

Algorithm:

personnel.takeMedicalHistory(patient: PERSON, chart: PATIENT_CHART)

Ensure:

chart contains medical history

measureBloodPressure: VOID

Purpose:

Measure patient's blood pressure using
bloodPressureGauge

Require:

patient not void
bloodPressureGauge not void
personnel.conforms_to(NON_CRITICAL_CARE_NURS
E)

Algorithm:

personnel.measureBloodPressure(patient: PERSON,
bloodPressureGauge: SPHYGMOMANOMETER,
chart: PATIENT_CHART)

Ensure:

chart contains blood pressure

drawBlood: VOID

Purpose:

Take a blood sample from patient

Require:

patient not void
bloodDrawingKit not void

bloodDrawingSupplies.ct > 0

personnel.conforms_to(PHLEBOTOMIST)

Algorithm:

bloodSample :=personnel.drawBlood(patient,
bloodDrawingKit, chart)

Ensure:

bloodSample not void;
chart updated

measureLDLCholesterol: VOID

Purpose:

measure the LDL cholestereol level of *patient*

Require:

patient not void

Algorithm:

drawBlood
send *bloodSample* to lab for analysis
wait for results
record results in chart

Ensure:

Results recorded in chart

measureHDLCholesterol: VOID

Purpose:

measure the HDL cholestereol level of *patient*

Require:

patient not void

Algorithm:

drawBlood

send *bloodSample* to lab for analysis

wait for results

record results in chart

Ensure:

Results recorded in chart

measureTotalCholesterol: VOID

Purpose:

measure the total cholesterol level of *patient*

Require:

patient not void

Algorithm:

drawBlood

send *bloodSample* to lab for analysis

wait for results

record results in chart

Ensure:

Results recorded in chart

performEKG: VOID

Purpose:

perform an EKG measurement on *patient*

Require:

patient not void

personnel not void

personnel.conformsTo(PHLEBOTOMIST) or

personnel.conformsTo(LOW_LEVEL_TECHNI

CIAN)

ekgMachine not void

Algorithm:

chart.insert(personnel.performEKG(patient,

ekgMachine))

Ensure:

chart.contains(ekg)

measureBloodPressure: <RESULT TYPE>

Purpose:

To be specified

measureProteinLevelInUrine: <RESULT TYPE>

Purpose:

To be specified

measureTemperature: <RESULT TYPE>

Purpose:

To be specified

measureWeight: <RESULT TYPE>

Purpose:

To be specified

takePersonalMedicalHistory: <RESULT TYPE>

Purpose:
To be specified

performHivAntiBodyTest: <RESULT TYPE>

Purpose:
To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:
To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:
To be specified

treatWithNiacin: <RESULT TYPE>

Purpose:
To be specified
Invariants:

Patient's chart present

patient not void implies *chart* not void and *chart.patient* =
patient

Comments:

Differentiating features

Presumably the differenting features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Connection to laboratory

In measureLDLCholesterol, how does the blood sample get to the lab? And to what lab?

Answer: measureLDLCholesterol in department layer sends it to lab. This may imply that there is no measureLDLCholesterol operation in th facility layer, since all that's done in the facility is to draw blood.

Class *MEDICAL_WARD*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:
Operations:
Invariants:

<invariant>

<description of class invariant>

Comments:

Differentiating features

Presumably the differentiating features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *NEONATAL_INTENSIVE_CARE_UNIT*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:
Operations:

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

Presumably the differentiating features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *NURSES_STATION*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

<role>: <TYPE>

<description of the component class and its role in the current class>

Operations:

<opr>: <RESULT TYPE>

Purpose:
<description of the purpose of the operation>
Require:
<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

Invariants:

<invariant>

<description of class invariant>

Comments:

Differentiating features

Presumably the differenting features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *OPERATING_ROOM*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:

To be specified

treat21TrisomyFetus: <RESULT TYPE>

Purpose:

To be specified

treatCesarean: <RESULT TYPE>

Purpose:

To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

Presumably the differenting features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *PARKING_LOT*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

<role>: <TYPE>

<description of the component class and its role in the current class>

Operations:

<opr>: <RESULT TYPE>

Purpose:

<description of the purpose of the operation>

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

Invariants:

<invariant>

<description of class invariant>

Comments:

Differentiating features

Presumably the differenting features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *PEDIATRIC_WARD*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

Operations:

Invariants:

<invariant>

<description of class invariant>

Comments:

Differentiating features

Presumably the differentiating features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *PHYSICIANS_OFFICE*

Description:

The general office for a physician to meet with patients. Doctors talk to patients in these offices, no examinations are done here.

Specializes:

FACILITY

The general facility.

Aggregates:

currentPatient: PERSON

The patient currently in the office.

doctor: PHYSICIAN

The physician using the office.

antiTobaccoHandouts:

SET[ANTI_TOBACCO_EDUCATION_HANDOUT]

Handouts for the anti-tobacco education.

dietHandouts:

SET[DIET_EDUCATION_HANDOUT]

Handouts for the diet education.

cleanNeedleHandouts:

SET[CLEAN_NEEDLE_EDUCATION_HANDOUT]

Handouts for the clean needle education.

sexHandouts: SET[SEX_EDUCATION_HANDOUT]

Handouts for the sex education.

prenatalHandouts:

SET[PRENATAL_EDUCATION_HANDOUT]

Handouts for the prenatal education.

equipment: SET[EQUIPMENT]

The standard physician's office equipment; see comment.
Operations:

talkToPatient

Purpose:

Doctor talks to currentPatient.

Require:
 currentPatient not void
 doctor not void

Algorithm:
 deferred

Ensure:
 <postconditions for the operation>

giveAntiTobaccoEducation: VOID

Purpose:
 Give anti-tobacco education to *currentPatient*.

Require:
 currentPatient not void
 doctor not void
 antiTobaccoHandouts.ct >= 1
 equipment /= void

Algorithm:
 doctor.giveAntiTobaccoEducation(currentPatient,
 antiTobaccoHandouts.item, equipment)

Ensure:
 <postconditions for the operation>

giveDietEducation: VOID

Purpose:
 Give diet education to *currentPatient*

Require:
 currentPatient not void
 doctor not void

dietHandouts.ct >= 1
equipment /= void

Algorithm:
 doctor.giveDietEducation(currentPatient,
 dietHandouts.item, equipment)

Ensure:
 students receive diet education

giveCleanNeedleEducation: VOID

Purpose:
 Give clean needle education to *currentPatient*

Require:
 currentPatient not void
 doctor not void
 dietHandouts.ct >= 1
 equipment /= void

Algorithm:
 doctor.giveCleanNeedleEducation(currentPatient,
 cleanNeedleHandouts.item, equipment)

Ensure:
 students receive clean needle education

giveSexEducation: VOID

Purpose:
 Give sex education to *currentPatient*

Require:
 currentPatient not void
 doctor not void

dietHandouts.ct >= 1
equipment != void

Algorithm:

doctor.giveSexEducation(*currentPatient*,
sexHandouts.item, *equipment*)

Ensure:

students receive diet education

givePrenatalEducation: VOID

Purpose:

Give prenatal education to *currentPatient*

Require:

currentPatient not void
doctor not void
dietHandouts.ct >= 1
equipment != void

Algorithm:

doctor.givePrenatalEducation(*currentPatient*,
prenatalHandouts.item, *equipment*)

Ensure:

students receive prenatal education

giveMaternalAntiTobaccoEducation: <RESULT TYPE>

Purpose:

To be specified

giveMaternalDrugEducation: <RESULT TYPE>

Purpose:

To be specified

giveMaternalHivEducation: <RESULT TYPE>

Purpose:

To be specified

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:

To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:

To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:

To be specified

treatWithNiacin: <RESULT TYPE>

Purpose:

To be specified

Invariants:

<invariant>

<description of class invariant>

Comments:

Equipment

What specific equipment should be in every physicians office?

giveAntiTobaccoEducation requirements

Can we be more specific about what equipment is required?

Handouts

see comment in *CLASS_ROOM*

Class RADIOLOGY

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

<role>: <TYPE>

<description of the component class and its role in the current class>

Operations:

<opr>: <RESULT TYPE>

Purpose:

<description of the purpose of the operation>

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

Invariants:

<invariant>

<description of class invariant>

Comments:

Differentiating features

Presumably the differing features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *STORAGE*Description:

<description of purpose and defining characteristics of the class>

Specializes:***FACILITY***

The general facility.

Aggregates:

<role>: <TYPE>

<description of the component class and its role in the current class>

Operations:

<opr>: <RESULT TYPE>

Purpose:

<description of the purpose of the operation>

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

Invariants:

<invariant>

<description of class invariant>

Comments:**Differentiating features**

Presumably the differenting features of these facilities are the interventions that can be performed there and the associated personnel, equipment, supplies and drugs. What are the specific features for this class?

Class *TELEMETRY_UNIT*Description:

A telemetry unit .

Specializes:***PATIENT_FACILITY***

The general facility for treating patients.

Aggregates:Operations:

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *TREADMILL_ROOM*

Description:

A room containing a treadmill

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:
Operations:

performTreadmillTest: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *ULTRASOUND_ROOM*

Description:

A room for performing ultrasound procedures.

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:
Operations:

measureFetusSizeWithUltrasound: <RESULT TYPE>

Purpose:
To be specified

measurePelvisSizeUltrasound: <RESULT TYPE>

Purpose:
To be specified

measure21TrisomyWithAmniocentesis: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *X_ROOM*

Description:

A room for taking x-rays

Specializes:

PATIENT_FACILITY

The general facility for treating patients.

Aggregates:

Operations:

measurePelvisSizeWithXRay: <RESULT TYPE>

Purpose:

To be specified

performAngiogram: <RESULT TYPE>

Purpose:

To be specified

performAngioplasty: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *WAITING_ROOM*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

FACILITY

The general facility.

Aggregates:

patients: BOUNDED_SETPERSON

The patients waiting in the waiting room. Bounded by the capacity of the waiting room.

Operations:

<opr>: <RESULT TYPE>

Purpose:

<description of the purpose of the operation>

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

Invariants:

bounded by capacity

patients.ub = capacity

No queue

queue void

Comments:

Shared waiting room

The waiting room may be shared by more than one department and by more than one service within a department. A waiting room is thus more than a single queue; it is several queues sharing the same storage.

DEPARTMENT CLUSTER

Abstract classes

Class *DEPARTMENT*

Description:

The general, abstract out-patient service facility. The basic organizational unit of the out-patient service.

Specializes:

FACILITY

The general facility

Typical usage

The waiting room is typically not used directly. Instead, references to it are given to various queues to use as storage. Patients are not directly put in the waiting room, they are inserted in queues.

Waiting room queue

WAITING_ROOM inherits a feature queue from REUSEABLE_RESOURCE. But since a waiting room is the storage for one or more queues, it doesn't make much sense for it to have a queue itself. Hence the invariant.

Immutable representations

Aggregates:

waitingArea: WAITING_ROOM

A place for patients to wait. May be shared with another department.

Operations:

Invariants:

Comments:

Specialization of *FACILITY*?

Is *DEPARTMENT* really a specialization of *FACILITY* or are the two synonymous?

Answer: Yes it is, the two are not the same. As indicated by the description, departments are associated with the out-patient service only.

Personnel

Does a department, as opposed to a specific room, have any personnel associated with it? Is a department purely a physical unit (i.e. the rooms) or it is an administrative unit as well. For instance, does it have a staff?

Role in interventions

The "Small Table of Interventions" [iii], lists a department for each intervention, but what role does the department play in delivering interventions?

Shared waiting room

The waiting room may be shared by more than one department and by more than one service within a department. A waiting room is thus more than a single queue; it is several queues sharing the same storage.

Class *APPOINTMENT_DEPARTMENT*

Description:

A department which has a receptionist and accepts appointments.

Specializes:

DEPARTMENT

The general department

Aggregates:

receptionArea: RECEPTION_AREA

A reception area.

receptionist: RECEPTIONIST

The receptionist for the department.
Operations:

makeAppointment(p: PERSON)

Purpose:

Makes an appointment for person p with the the current department.

Require:

Algorithm:
deferred

Ensure:

The appointment is made by the *receptionist* or the *generalAppointmentMaker*

Invariants:

receptionist's queue storage

receptionist.queue.storage = waitingArea

Comments:

General appointment maker.

Who or what is the general appointment maker? (ref: Mariano, notes from discussion with LenS, regarding [ii])

Class *MEDICAL_DEPARTMENT*

Description:

A department with facilities for physicians.

Specializes:

APPOINTMENT_DEPARTMENT

The general department that can accept appointments.

Aggregates:

physiciansOffices: SET[PHYSICIAN_OFFICE]

The collection of offices for physicians.

examinationRooms: SET[EXAMINATION_ROOM]

The collection of examination rooms in the department. Typically two for each physician's office.

examinationRoomsQueue: QUEUE[PERSON]

The queue of patients waiting for an examination room.

highLevelProcedureRooms:
SET[HIGH_LEVEL_PROCEDURE_ROOM]

The collection of high level procedure rooms in the department.

highLevelProcedureRoomsQueue:
QUEUE[PERSON]

The queue of patients waiting for a high level procedure room.

lowLevelProcedureRooms:
SET[LOW_LEVEL_PROCEDURE_ROOM]

The collection of low level procedure rooms in the department.

lowLevelProcedureRoomsQueue: QUEUE[PERSON]

The queue of patients waiting for a low level procedure room.

nursesStation: NURSES_STATION

Every medical department has a nurses station.

Operations:

drawBlood: <RESULT TYPE>

Purpose:
To be specified

measureBloodPressure: <RESULT TYPE>

Purpose:
To be specified

takePersonalMedicalHistory: <RESULT TYPE>

Purpose:
To be specified

receivePatient(p: PERSON): VOID

Purpose:
Check patient in for processing.

Require:
p not void

Algorithm:
receptionist.queue.insert(p)

Ensure:

Invariants:

shared queue for examinationRooms

for each room *r* in *examinationRooms*, *r.queue* = *examinationRoomsQueue*

shared queue for lowLevelProcedureRooms

for each room *r* in *lowLevelProcedureRooms*, *r.queue* = *lowLevelProcedureRoomsQueue*

shared queue for highLevelProcedureRooms

for some queue *qh* and for each room *r* in *highLevelProcedureRooms*, *r.queue* = *highLevelProcedureRoomsQueue*

shared storage for all queues

examinationRoomsQueue.storage = *lowLevelProcedureRoomsQueue.storage* = *highLevelProcedureRoomsQueue.storage* = *waitingArea*

Comments:

Operations

What operations can the general medical department do?

Receptionist's queue and receivePatient

receivePatient appears to just insert the patient in a queue. However, the receptionist is concurrently processing

patients off the queue. See RECEPTIONIST. But how does the receptionist get initialized?

Multiple queues

MEDICAL_DEPARTMENT inherits a queue from REUSEABLE_RESOURCE via DEPARTMENT and FACILITY. It does use this one and declares several more. Should the general reusable resource have several queues? Or is the queues associated with the department as a resource some other queue, e.g. the queue of patients waiting for appointments? The latter seems most likely.

Class *CVD_DEPARTMENT*

Description:

The general, abstract department that can process CVD related appointments

Specializes:

MEDICAL_DEPARTMENT

The general department with facilities for physicians

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Mutable representations

Class *CARDIAC_CARE_UNIT*

Description:

A cardiac care unit.

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:

To be specified

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CARDIOLOGY_DEPARTMENT*

Description:

The general cardiology department.

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

performAngiogram: <RESULT TYPE>

Purpose:

To be specified

performAngioplasty: <RESULT TYPE>

Purpose:

To be specified

performBypassSurgery: <RESULT TYPE>

Purpose:

To be specified

performEKG: <RESULT TYPE>

Purpose:

To be specified

performTreadmillTest: <RESULT TYPE>

Purpose:

To be specified

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:

To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:

To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:

To be specified

treatWithNiacin: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **EMERGENCY_DEPARTMENT**

Description:

An emergency medicine department

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *HIV_DEPARTMENT*

Description:

An HIV department

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

treatCMV: <RESULT TYPE>

Purpose:

To be specified

treatPCP: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class

OBSTETRICS_AND_GYNECOLOGY_DEPARTMENT

Description:

The general obstetrics and gynecology department.

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:
Operations:

<i>measure21TrisomyWithAmniocentesis:</i> <i><RESULT TYPE></i>	<i>treatCesarean:</i> <i><RESULT TYPE></i>
Purpose: To be specified	Purpose: To be specified
<i>measureBloodOxygenAndNutrientsInPlacenta:</i> <i><RESULT TYPE></i>	<i>treatEclampsia:</i> <i><RESULT TYPE></i>
Purpose: To be specified	Purpose: To be specified
<i>measureFetusSizeWithUltrasound:</i> <i><RESULT TYPE></i>	<i>treatHIV+Fetus:</i> <i><RESULT TYPE></i>
Purpose: To be specified	Purpose: To be specified
<i>measureProteinLevelInUrine:</i> <i><RESULT TYPE></i>	<i>treatLowBirthWeightFetus:</i> <i><RESULT TYPE></i>
Purpose: To be specified	Purpose: To be specified
<i>treat21TrisomyFetus:</i> <i><RESULT TYPE></i>	<i>treatMaternalHIV:</i> <i><RESULT TYPE></i>
Purpose: To be specified	Purpose: To be specified
	<i>treatNormalTermFetus:</i> <i><RESULT TYPE></i>
	Purpose: To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PATIENT_EDUCATION_DEPARTMENT*

Description:

The general patient education department.

Specializes:

DEPARTMENT

The general department.

Rename waitingArea to lobby; redefine lobby

Aggregates:

lobby: LOBBY

A lobby area.

classrooms: CLASS_ROOM

The collection of class rooms in the department.

Operations:

distributeCleanNeedles: <RESULT TYPE>

Purpose:
To be specified

distributeCondoms: <RESULT TYPE>

Purpose:
To be specified

giveAntiTobaccoEducation: <RESULT TYPE>

Purpose:
To be specified

giveCleanNeedleEducation: <RESULT TYPE>

Purpose:
To be specified

giveDietEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalAntiTobaccoEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalDrugEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalHivEducation: <RESULT TYPE>

Purpose:
To be specified

givePrenatalEducation: <RESULT TYPE>

Purpose:
To be specified

giveSexEducation: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Instructors

Should a education department have a collection of instructors associated with it? See also the question about personnel under DEPARTMENT

Class *PEDIATRICS_DEPARTMENT*

Description:

The general pediatrics department.

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PHARMACY_DEPARTMENT*

Description:

The general pharmacy department.

Specializes:

DEPARTMENT

The general department.

Aggregates:

pharmacists: SET[PHARMACIST]

The pharmacists working in the pharmacy.

stock: SET[DRUG]

The pharmacy's supply of drugs.

Operations:

fillPrescription(rx: PRESCRIPTION): DRUG

Purpose:

Fills the prescription *rx*.

Require:

not rx.void

Algorithm:

deferred

Ensure:

result matches prescription

Invariants:

Comments:

Differentiating features?

What are the specific differentiating features? Do the features I've included make sense?

Class *PRIMARY_CARE_DEPARTMENT*

Description:

The general primary care department.

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

distributeCleanNeedles: <RESULT TYPE>

Purpose:

To be specified

distributeCondoms: <RESULT TYPE>

Purpose:

To be specified

giveAntiTobaccoEducation: <RESULT TYPE>

Purpose:

To be specified

giveCleanNeedleEducation: <RESULT TYPE>

Purpose:

To be specified

giveDietEducation: <RESULT TYPE>

Purpose:

To be specified

<i>giveMaternalAntiTobaccoEducation:</i> <RESULT TYPE>	<i>measureLdlCholesterol:</i> <RESULT TYPE>
Purpose: To be specified	Purpose: To be specified
<i>giveMaternalDrugEducation:</i> <RESULT TYPE>	<i>measureTemperature:</i> <RESULT TYPE>
Purpose: To be specified	Purpose: To be specified
<i>giveMaternalHivEducation:</i> <RESULT TYPE>	<i>measureTotalCholesterol:</i> <RESULT TYPE>
Purpose: To be specified	Purpose: To be specified
<i>givePrenatalEducation:</i> <RESULT TYPE>	<i>measureWeight:</i> <RESULT TYPE>
Purpose: To be specified	Purpose: To be specified
<i>giveSexEducation:</i> <RESULT TYPE>	<i>performCd4TCellTest:</i> <RESULT TYPE>
Purpose: To be specified	Purpose: To be specified
<i>measureHdlCholesterol:</i> <RESULT TYPE>	<i>performCMVTest:</i> <RESULT TYPE>
Purpose: To be specified	Purpose: To be specified

performEKG: <RESULT TYPE>

Purpose:
To be specified

performFungusTest: <RESULT TYPE>

Purpose:
To be specified

performFungusTest: <RESULT TYPE>

Purpose:
To be specified

performHivAntiBodyTest: <RESULT TYPE>

Purpose:
To be specified

performHivPerTest: <RESULT TYPE>

Purpose:
To be specified

performLungOxygenTest: <RESULT TYPE>

Purpose:
To be specified

performLymphNodeSizeTest: <RESULT TYPE>

Purpose:
To be specified

performPCPTest: <RESULT TYPE>

Purpose:
To be specified

performTreadmillTest: <RESULT TYPE>

Purpose:
To be specified

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:
To be specified

treatCMV: <RESULT TYPE>

Purpose:
To be specified

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified

treatMaternalHIV: <RESULT TYPE>

Purpose:
To be specified

treatPCP: <RESULT TYPE>

Purpose:
To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:
To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:
To be specified

treatWithNiacin: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *RADIOLOGY_DEPARTMENT*

Description:

The general pediatrics department.

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

measurePelvisSizeUltrasound: <RESULT TYPE>

Purpose:
To be specified

measurePelvisSizeWithXRay: <RESULT TYPE>

Purpose:
To be specified

performAngiogram: <RESULT TYPE>

Purpose:
To be specified

performAngioplasty: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *SAME_DAY_APPOINTMENT_DEPARTMENT*

Description:

The same day appointment department.

Specializes:

APPOINTMENT_DEPARTMENT

The general appointment department

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

PERSONNEL CLUSTER

Abstract classes

Class *PERSONNEL*

Description:

The general, abstract personnel

Specializes:

PERSON

the general, abstract person.

Class *TELEMETRY_UNIT*

Description:

A telemetry unit

Specializes:

MEDICAL_DEPARTMENT

The general medical department

Aggregates:

Operations:

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Immutable representations

REUSEABLE_RESOURCE

The general reusable, schedulable resource.

Aggregates:

qualifications: SET[PROCEDURE]

The collection of procedures the personnel is allowed to do.

employmentHistory: TIME

The experience, length of time of service, of the personnel.

absenteeism: REAL

The probability of the personnel being absent.

behaviors: <TYPE?>

The propensity of the physician to use interventions or to adopt clinical or management policies. See comment below.

recruitCost: MONEY

The cost to recruit the personnel

trainCost: MONEY

The cost to *train* the personnel

salaryCost: MONEY

The base salary of the personnel

overtimeCost: MONEY

The cost for overtime for the personnel

retirementCost: MONEY

The cost of the retirement plan for the personnel
Operations:

skillLevel(i: PROCEDURE, f: FACILITY): REAL

Purpose:

A measure of the physicians skill at performing intervention i at facility f

Require:

qualifications.contains(i)

Algorithm:

deferred

Ensure:

<postconditions for the operation>

Invariants:

Comments:

Skill level

The skill level depends on the intervention and location in principle. However, this requires that the physician object know about all interventions and facilities. It may be more desirable, from a maintenance point of view, to have the facility where the intervention is performed know about physicians and interventions, since this is necessary anyway.

Skill level and qualifications

One approach would be to define a type QUALIFICATION, which would contain a procedure code and a skill level. We would then have *qualifications*: SET[QUALIFICATION] which would represent both the personnel qualification to do a certain procedure and thier skill at it. The skill levels would be generated when the personnel object was created by sampling some distribution.

Behaviors

It is not yet clear how to model behaviors. The current notion is some combination of rules + distributions, similar to the genetic predisposition mechanism planned for diseases.

Specialists

Each specialization has specific behavior and can perform specific procedures. In principle, we need to program each specialization as a separate class. But is there some acceptable parameterization (classification formally) that would allow us to represent all these as a single class?

Interventions and procedures

I've put almost all the education interventions here, but they probably belong in one or more specializations that have yet to be identified.

Mutable representations

Class ADMINSTRATOR

Description:

<description of purpose and defining characteristics of the class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is allowed to perform. What are the specific procedures for this class?

Class CARDIAC_CARE_UNIT_NURSE

Description:

A nurse in a cardiac care unit

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:
Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type of nurse, or just a role?

Class *CARDIAC_CARE_UNIT_STAFF*

Description:

Personnel in the cardiac_care_unit

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:
Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type of personnel, or just a role?

Class *CARDIAC_NURSE*

Description:

A cardiac nurse

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type or nurse, or just a role? Is it the same as *CARDIAC_CARE_UNIT_NURSE*?

Class *CLERK*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is allowed to perform. What are the specific procedures for this class?

Class *CRITICAL_CARE_NURSE*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is allowed to perform. What are the specific procedures for this class?

Class *DELIVERY_ROOM_PERSONNEL*

Description:

Personnel in the delivery room

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:

To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type of personnel, or just a role?

Class *DIETICIAN*

Description:

A dietician

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

giveDietEducation(students:

handouts:

SET[PERSON],

SET[DIET_EDUCATION_HANDOUTS],

classroomEquipment:

SET[EQUIPMENT]): VOID

Purpose:

Givece diet education to *students* using *handouts* and *equipment*

Require:

students, handouts, equipment not void
handouts.ct >= students.ct

Algorithm:

deferred

Ensure:

students receive diet education

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *HIGH_LEVEL_TECHNICIAN*

Description:

<description of purpose and defining characteristics of the
class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

measure21TrisomyWithAmniocentesis: <RESULT
TYPE>

Purpose:

To be specified

measureFetusSizeWithUltrasound: <RESULT
TYPE>

Purpose:

To be specified

performTreadmillTest: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is

allowed to perform. What are the specific procedures for this class?

Class INPATIENT_PERSONNEL

Description:

Personnel attending to inpatients

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

treat21TrisomyFetus: <RESULT TYPE>

Purpose:

To be specified

treatCesarean: <RESULT TYPE>

Purpose:

To be specified

treatCMV: <RESULT TYPE>

Purpose:

To be specified

treatEclampsia: <RESULT TYPE>

Purpose:

To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:
To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
To be specified

treatPCP: <RESULT TYPE>

Purpose:
To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type of personnel, or just a role?

Class **INSTRUCTOR**

Description:

An instructor for patient education

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

*giveAntiTobaccoEducation(students: SET[PERSON],
handouts:
SET[ANTI_TOBACCO_HANDOUTS],
classRoomEquipment:
SET[EQUIPMENT]): VOID*

Purpose:

Give anti-tobacco education to students using handouts
and equipment.

Require:

*students, handouts and classRoomEquipment not void
handouts.ct >= students.ct*

Algorithm:

deferred

Ensure:

students receive education

*giveCleanNeedleEducation(students: SET[PERSON],
handouts:
SET[CLEAN_NEEDLE_HANDOUTS],
classRoomEquipment:
SET[EQUIPMENT]): VOID*

Purpose:
Give clean needle education to students using handouts
and equipment.

Require:
*students, handouts and classRoomEquipment not void
handouts.ct >= students.ct*

Algorithm:
deferred

Ensure:
students receive education

*giveSexEducation(students: SET[PERSON],
handouts: SET[SEX_HANDOUTS],
classRoomEquipment:
SET[EQUIPMENT]): VOID*

Purpose:
Give sex education to students using handouts and
equipment.

Require:
*students, handouts and classRoomEquipment not void
handouts.ct >= students.ct*

Algorithm:
deferred

Ensure:
students receive education

*givePrenatalEducation(students: SET[PERSON],
handouts:
SET[PRENATAL_HANDOUTS],
classRoomEquipment:
SET[EQUIPMENT]): VOID*

Purpose:
Give prenatal education to students using handouts and
equipment.

Require:
*students, handouts and classRoomEquipment not void
handouts.ct >= students.ct*

Algorithm:
deferred

Ensure:
students receive education

*giveMaternalAntiTobaccoEducation: <RESULT
TYPE>*

Purpose:
To be specified

giveMaternalDrugEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalHivEducation: <RESULT TYPE>

Purpose:
To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:
To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type of personnel, or just a role?

Class *LAB_TECHNICIAN*
Description:

Laboratory personnel

Specializes:
Aggregates:
Operations:

measureLDLCholesterol(sample: blood_sample, analyzer: EQUIPMENT): VOID

Purpose:
measure the LDL cholesterol level of *sample* using *analyzer*.

Require:
sample not void

Algorithm:
deferred

Ensure:
<postconditions for the operation>

measureHDLCholesterol(sample: blood_sample, analyzer: EQUIPMENT): VOID

Purpose:
measure the HDL cholesterol level of *sample* using *analyzer*.

Require:
sample not void

Algorithm:
deferred

Ensure:
<postconditions for the operation>

*measureTotalCholesterol(sample: blood_sample,
analyzer: EQUIPMENT): VOID*

Purpose:

measure the total cholesterol level of sample using
analyzer.

Require:

sample not void

Algorithm:

deferred

Ensure:

<postconditions for the operation>

performCd4TCellTest: <RESULT TYPE>

Purpose:

To be specified

performCMVTest: <RESULT TYPE>

Purpose:

To be specified

performFungusTest: <RESULT TYPE>

Purpose:

To be specified

performFungusTest: <RESULT TYPE>

Purpose:

To be specified

performHivPcrTest: <RESULT TYPE>

Purpose:

To be specified

performLungOxygenTest: <RESULT TYPE>

Purpose:

To be specified

performPCPTest: <RESULT TYPE>

Purpose:

To be specified

*measure21TrisomyWithAmniocentesis: <RESULT
TYPE>*

Purpose:

To be specified

*measureBloodOxygenAndNutrientsInPlacenta:
<RESULT TYPE>*

Purpose:

To be specified

measureProteinLevelInUrine: <RESULT TYPE>

Purpose:
To be specified

performHivAntiBodyTest: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Equipment for cholesterol tests

Almost certainly, *analyzer* should be some specialization of *EQUIPMENT*.

Class LOGISTICAL_SUPPORT

Description:

<description of purpose and defining characteristics of the class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:
Operations:
Invariants:
Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is allowed to perform. What are the specific procedures for this class?

Class LOW_LEVEL_TECHNICIAN

Description:

A low level technician
Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:
Operations:

performEKG(patient: PERSON, ekgMachine: EKG_MACHINE, chart: PATIENT_CHART): VOID

Purpose:

Perform an EKG measurement of *patient* using *ekgMachine* and record in *chart*

Require:

patient not void

ekgMachine not void
chart not void

Algorithm:

ekg := *ekgMachine.measure(patient)*
chart.insert(ekg)

Ensure:

chart.contains(ekg)

distributeCleanNeedles: <*RESULT TYPE*>

Purpose:

To be specified

distributeCondoms: <*RESULT TYPE*>

Purpose:

To be specified

Invariants:

Comments:

Class

NEONATAL_INTENSIVE_CARE_UNIT_PERSONNEL

Description:

Personnel in a neonatal intensive care unit.

Specializes:

Aggregates:

Operations:

treatLowBirthWeightFetus: <*RESULT TYPE*>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type of personnel, or just a role?

Class *NON_MEDICAL_SUPPORT*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is allowed to perform. What are the specific procedures for this class?

Class *NURSE*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

measure21TrisomyWithAmniocentesis: <RESULT
TYPE>

Purpose:
To be specified

measureBloodPressure: <RESULT TYPE>

Purpose:
To be specified

measurePelvisSizeUltrasound: <RESULT TYPE>

Purpose:
To be specified

measurePelvisSizeUltrasound: <RESULT TYPE>

Purpose:
To be specified

measureProteinLevelInUrine: <RESULT TYPE>

Purpose:
To be specified

measureWeight: <RESULT TYPE>

Purpose:
To be specified

performAngiogram: <RESULT TYPE>

Purpose:
To be specified

performAngioplasty: <RESULT TYPE>

Purpose:
To be specified

takePersonalMedicalHistory: <RESULT TYPE>

Purpose:
To be specified

treat21TrisomyFetus: <RESULT TYPE>

Purpose:
To be specified

treatCMV: <RESULT TYPE>

Purpose:
To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:
To be specified

treatPCP: <RESULT TYPE>

Purpose:
To be specified

measureWeight: <RESULT TYPE>

Purpose:
To be specified

Invariants:

Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is allowed to perform. What are the specific procedures for this class?

Class *NURSE_PRACTITIONER*

Description:

A nurse practitioner

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

measureProteinLevelInUrine: <RESULT TYPE>

Purpose:
To be specified

measureTemperature: <RESULT TYPE>

Purpose:
To be specified

measureTemperature: <RESULT TYPE>

Purpose:
To be specified

performLymphNodeSizeTest: <RESULT TYPE>

Purpose:
To be specified

treatWithAspirin: <RESULT TYPE>

Purpose:
To be specified

treatWithLovastatin: <RESULT TYPE>

Purpose:
To be specified

treatWithNiacin: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *OPERATING_ROOM_PERSONNEL*

Description:

Personnel in an operating room
Specializes:

PERSONNEL

The general non-physician employee.
Aggregates:
Operations:

treat21TrisomyFetus: <RESULT TYPE>

Purpose:
To be specified

treatCesarean: <RESULT TYPE>

Purpose:
To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PHARMACIST*

Description:

A pharmacist.
Specializes:

PERSONNEL

The general non-physician employee.
Aggregates:
Operations:

dispenseNiacin: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

Presumably this class is specialized by defining operations which correspond to the procedures the personnel type is allowed to perform. What are the specific procedures for this class?

Class *PHLEBOTOMIST*

Description:

A person who draws blood.

Specializes:

PERSONNEL

The general non-physician personnel.

Aggregates:

Operations:

drawBlood(patient: PERSON, kit: BLOOD_DRAWING_KIT, chart: PATIENT_CHART): BLOOD_SAMPLE

Purpose:

draw a blood sample from *patient* using *kit* and record in *chart*.

Require:

patient not void
kit not void
chart not void

Algorithm:
deferred

Ensure:
result not void

performEKG: <RESULT TYPE>

Purpose:
To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *RECEPTIONIST*

Description:

A departmental receptionist.

Specializes:

PERSONNEL

The general personnel

Aggregates:**department: APPOINTMENT_DEPARTMENT**<description of the component class and its role in the
current class>Operations:**checkPatientIn: VOID**

Purpose:

Checks in patient for processing.

Require:

<preconditions of the operation>

Algorithm:

```

patient := current.queue.remove
request Kaiser ID card
verify group coverage
verify appointment and determine urgency
if receiving for multiple departments
    identify proper department

```

select *cpr*: CLINICAL_PROCESSING_RECORD

collect co-payment registration fee

give *cpr* to *patient*

```

department.lowLevelProcedureRoomQueue.insert(patient)

```

Ensure:

<postconditions for the operation>

receivePatients: VOID

Purpose:

The daily routine for the receptionist.

Require:

<preconditions of the operation>

Algorithm:

```

until quitting time
{
    if not queue.empty
        checkPatientIn
    else
        wait until not queue.empty
    endif
}

```

Ensure:

<postconditions for the operation>

Invariants:Comments:**Differentiating features**

What are the specific differentiating features of this class?

Class SURGICAL_STAFFDescription:<description of purpose and defining characteristics of the
class>

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Sames as *OPERATING_ROOM_PERSONNEL*?

What's the difference between these two classes

Class *TELEMETRY_UNIT_NURSE*

Description:

A nurse in a telemetry unit.

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Subtype or role?

Is this really a special type of personnel, or just a role?

Class *TELEMETRY_UNIT_STAFF*

Description:

Personnel in a telemetry unit

Specializes:

PERSONNEL

The general non-physician employee.

Aggregates:

Operations:

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?
EQUIPMENT CLUSTER

Abstract classes

Class *EQUIPMENT*

Description:

The general, abstract medical equipment.

Specializes:

REUSEABLE_RESOURCE

The general reusable resource.

Aggregates:

identifier: <TYPE?>

A unique identifier for the equipment; e.g. a serial number.
See comment below

isMobile: BOOLEAN

True if equipment can be moved.

age: INTEGER

Numer of uses to date

Subtype or role?

Is this really a special type of personnel, or just a role?
Immutable representations

lifeExpectancy: INTEGER

Expected total number of uses.

reliability: REAL

Probability of failure.

Operations:

<opr>: <RESULT TYPE>

Purpose:

<description of the purpose of the operation>

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

Invariants:

<invariant>

<description of class invariant>

Comments:

Identifiers

What type of thing is this? How do we represent it?

Generalizing age, lifeExpectancy and reliability

Can't these three be generalized and moved to REUSEABLE_RESOURCE? Don't all reusable resources have some version of these three? In fact, can't reliability be moved to RESOURCE?

Other resource requirements

KDA5.2 has a discussion of "other resource requirements" for EQUIPMENT. It would seem that these are place holders for features of specialized equipment. See KDA5.2

Other specializations

Specializations of EQUIPMENT are listed below. What other specializations do we need to specify?

Class *DIAGNOSTIC_EQUIPMENT*

Description:

The general diagnostic equipment - something that can measure something about a patient

Specializes:

EQUIPMENT

The general equipment

Aggregates:

Operations:

measure(patient: PERSON): RECORD

Purpose:

Measures patient in some way

Require:

patient not void

Algorithm:

deferred

Ensure:

result not void

Invariants:

Comments:

Additional features

What else do we need here?

Mutable representations

Class *ABORTION_EQUIPMENT*

Description:

Equipment used for performing abortions.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

treat21TrisomyFetus: <RESULT TYPE>

Purpose:

To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *AMNIOCENTESIS_EQUIPMENT*

Description:

Equipment used for performing an amniocentesis test.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

measure21TrisomyWithAmniocentesis: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *ANGIOGRAM_EQUIPMENT*

Description:

Equipment used for taking an angiogram

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

performAngioplasty: <RESULT TYPE>

Purpose:

To be specified

performAngiogram: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class ANGIOPLASTY_EQUIPMENT

Description:
Equipment used for performing an angioplasty operation.
Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

performAngioplasty: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class BLOOD_DRAWING_KIT

Description:
Collection of equipment needed to draw blood.
Specializes:

EQUIPMENT

The general equipment.

Aggregates:
Operations:

drawBlood: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

EQUIPMENT or SUPPLIES?

Is this class a specialization of *EQUIPMENT* or
SUPPLIES?

Class

BLOOD_OXYGEN_AND_NUTRIENT_MEASURING_EQUIPMENT

Description:
Equipment for measuring blood oxygen etc.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

measureBloodOxygenAndNutrientsInPlacenta:
<RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CARDIAC_BYPASS_EQUIPMENT*

Description:

Equipment used for performing caridac bypass surgery.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CARDIAC_CARE_UNIT_EQUIPMENT*

Description:

Equipment used in a cardiac care unit

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CD4_TCELL_TEST_ANALYSIS_EQUIPMENT*

Description:

Equipment used to analyze results from cd4 t-cell tests.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

performCd4TCellTest: <RESULT TYPE>

Purpose:
To be specified

performCMVTest: <RESULT TYPE>

Purpose:
To be specified

performFungusTest: <RESULT TYPE>

Purpose:
To be specified

performFungusTest: <RESULT TYPE>

Purpose:
To be specified

performLungOxygenTest: <RESULT TYPE>

Purpose:
To be specified

performPCPTTest: <RESULT TYPE>

Purpose:
To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CLASS_ROOM_EQUIPMENT*

Description:

Equipment used in a class room

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

giveAntiTobaccoEducation: <RESULT TYPE>

Purpose:
To be specified

giveCleanNeedleEducation: <RESULT TYPE>

Purpose:
To be specified

giveDietEducation: <RESULT TYPE>

Purpose:
To be specified

*giveMaternalAntiTobaccoEducation:
TYPE>*

Purpose:
To be specified

giveMaternalDrugEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalHivEducation: <RESULT TYPE>

Purpose:
To be specified

givePrenatalEducation: <RESULT TYPE>

Purpose:
To be specified

giveSexEducation: <RESULT TYPE>

Purpose:
To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *DELIVERY_ROOM_EQUIPMENT*

Description:

Equipment used in a delivery room.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *ELISA_EQUIPMENT*

Description:

Equipment for performing an enzyme linked immunosorbant assay (ELISA) test
Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

performHivAntiBodyTest: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *EKG_MACHINE*

Description:

A machine for measuring a persons EKG
Specializes:

DIAGNOSTIC_EQUIPMENT

The general equipment for measuring patients

Redefine *measure*

Aggregates:
Operations:

measure(patient: PERSON): EKG_RECORD

Purpose:
Measures *patient's* EKG
Require:
patient not void

Algorithm:
deferred

Ensure:
 result not void

Invariants:
Comments:

Additional features

What else do we need here?

Class *HIV_ANTI_BODY_TEST_ANALYSIS_EQUIPMENT*
Description:

Equipment for performing an HIV anti-body test analysis.
Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

performHivAntiBodyTest: <RESULT TYPE>

Purpose:
 To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *INPATIENT_EQUIPMENT*

Description:

Equipment used for treating inpatients.

Specializes:

Aggregates:

Operations:

treatCesarean: <RESULT TYPE>

Purpose:
 To be specified

treatCMV: <RESULT TYPE>

Purpose:
 To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
 To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
 To be specified

• *treatPCP: <RESULT TYPE>*

Purpose:
 To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *INTENSIVE_CARE_UNIT_EQUIPMENT*

Description:

Equipment used in an intensive care unit.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *MI_ER_PACKAGE*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

Presumably each specialized type of equipment defines operations corresponding to its specialized functions and the interventions it participates in. What are the specific features for this class?

Class

NEONATAL_INTENSIVE_CARE_UNIT_EQUIPMENT

Description:

Equipment used in a neo-natal intensive care unit.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

Operations:
Invariants:
Comments:

Differentiating features

Presumably each specialized type of equipment defines operations corresponding to its specialized functions and the interventions it participates in. What are the specific features for this class?

Class *PHYSICIANS_OFFICE_EQUIPMENT*

Description:
Equipment used in a physicians office
Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

giveAntiTobaccoEducation: <RESULT TYPE>

Purpose:
To be specified

giveCleanNeedleEducation: <RESULT TYPE>

Purpose:
To be specified

giveDietEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalAntiTobaccoEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalDrugEducation: <RESULT TYPE>

Purpose:
To be specified

giveMaternalHivEducation: <RESULT TYPE>

Purpose:
To be specified

givePrenatalEducation: <RESULT TYPE>

Purpose:
To be specified

giveSexEducation: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *OPERATING_ROOM_EQUIPMENT*

Description:

Equipment used in an operating room.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:

To be specified

treatCesarean: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PCR_ANALYSIS_EQUIPMENT*

Description:

Equipment for analyzing the results of an HIV PCR test.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

performHivPcrTest: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PCR_TEST_EQUIPMENT*

Description:

Equipment for performing an HIV PCR test.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

performHivPcrTest: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *SCALE*

Description:

Equipment for measuring weight.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

measureWeight: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *SHYGMOMANOMETER*

Description:

A device for measuring blood pressure.

Specializes:

DIAGNOSTIC_EQUIPMENT

The general diagnostic equipment

Redefine measure

Aggregates:

Operations:

measure(patient:

BLOOD_PRESSURE_RECORD

PERSON):

Purpose:

Measures patient in some way

Require:

patient not void

Algorithm:

deferred

Ensure:

result not void

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?
Class *TELEMETRY_UNIT_EQUIPMENT*
Description:

Equipment found in a telemetry unit
Specializes:

EQUIPMENT

The general equipment class.
Aggregates:
Operations:

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?
Class *THERMOMETER*
Description:

Equipment for measuring temperature.

Specializes:

EQUIPMENT

The general equipment class.
Aggregates:
Operations:

measureTemperature: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?
Class *THROMBOLYTIC_ADMINISTERING_EQUIPMENT*
Description:

Equipment for administering thrombolytics.
Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *TREADMILL_EKG*

Description:

A treadmill EKG

Specializes:

EKG_MACHINE

the general EKG machine

Aggregates:
Operations:

performTreadmillTest: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific features for this class?

treadmill test

Is this a different test, or just a different implementation of the general EKG test inherited from *EKG_MACHINE*?

Class *ULTRASOUND_EQUIPMENT*

Description:

Equipment for taking ultrasound measurements.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

measure21TrisomyWithAmniocentesis: <RESULT TYPE>

Purpose:
To be specified

measureFetusSizeWithUltrasound: *<RESULT TYPE>*

Purpose:
To be specified

measurePelvisSizeUltrasound: *<RESULT TYPE>*

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *URINE_SAMPLE_ANALYSIS_EQUIPMENT*

Description:

Equipment for analyzing urine samples.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

measureProteinLevelInUrine: *<RESULT TYPE>*

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *WESTERN_BLOT_TEST_EQUIPMENT*

Description:

Equipment for performing a western blot test.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:
Operations:

performHivAntiBodyTest: *<RESULT TYPE>*

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *X_RAY_EQUIPMENT*

Description:

Equipment for taking x-ray measurements.

Specializes:

EQUIPMENT

The general equipment class.

Aggregates:

Operations:

performAngiogram: <RESULT TYPE>

Purpose:

To be specified

SUPPLY CLUSTER

Abstract classes

Class *SUPPLY*

Description:

The general, abstract supply

Specializes:

RESOURCE

The general resource.

Aggregates:

reliability: REAL

The probability of failing to perform as expected.

performAngioplasty: <RESULT TYPE>

Purpose:

To be specified

measurePelvisSizeWithXRay: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?
Immutable representations

storageRequirements: <TYPE?>

See comment below

disposalRequirements: <TYPE?>

See comment below.

Operations:

Invariants:

Comments:

storage and disposal requirements

How do we represent these? How do we expect to use them?

Types of supplies

What types of supplies besides sponges do we need? Do we really need to differentiate or can we just have one class with some type code? Are there any differentiating features?

Mutable representations

Class *ABORTION_SUPPLIES*

Description:

Supplies used to perform an abortion

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

treat21TrisomyFetus: <RESULT TYPE>

Purpose:

To be specified

treatHIV+Fetus: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *AMNIOCENTESIS_SUPPLIES*

Description:

Supplies needed to perform an amniocentesis test.

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

measure21TrisomyWithAmniocentesis: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *ANGIOPLASTY_SUPPLIES*

Description:

Supplies used in angioplasty

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

performAngioplasty: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **ANTI_TOBACCO_EDUCATION_HANDOUT**

Description:

Handouts for anti-tobacco education.

Specializes:

Aggregates:

Operations:

giveAntiTobaccoEducation: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **BLOOD_DRAWING_SUPPLIES**

Description:

Supplies used for drawing blood.

Specializes:

Aggregates:

Operations:

drawBlood: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **BLOOD_OXYGEN_AND_NUTRIENT_MEASURING_SUPPLIES**

Description:

Supplies used in measuring blood oxygen, etc

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

measureBloodOxygenAndNutrientsInPlacenta:
<RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **BLOOD_SAMPLE**

Description:

A blood sample.

Specializes:

SUPPLY

The general non-reuseable resource.

Aggregates:

origin: DEPARTMENT

The department that drew the sample. The place to return lab tests to.

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Origin

Is this feature reasonable? How is information really handled?

Class **CARDIAC_BYPASS_SURGERY_SUPPLIES**

Description:

Supplies used in performing cardiac bypass surgery

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CD4_TCELL_TEST_ANALYSIS_SUPPLIES*

Description:

Supplies used in analyzing cd4 tcell test results

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

performCd4TCellTest: <RESULT TYPE>

Purpose:
To be specified

performCMVTest: <RESULT TYPE>

Purpose:
To be specified

performFungusTest: <RESULT TYPE>

Purpose:
To be specified

performFungusTest: <RESULT TYPE>

Purpose:
To be specified

performLungOxygenTest: <RESULT TYPE>

Purpose:
To be specified

performPCPTTest: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CHOLESTEROL_TEST_SUPPLIES*

Description:

Supplies used for testing blood for cholesterol.

Specializes:

Aggregates:

Operations:

measureHdlCholesterol: <RESULT TYPE>

Purpose:

To be specified

measureLdlCholesterol: <RESULT TYPE>

Purpose:

To be specified

measureTotalCholesterol: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CLEAN_NEEDLE_EDUCATION_HANDOUT*

Description:

Handouts for clean_needle education.

Specializes:

Aggregates:

Operations:

giveCleanNeedleEducation: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CONDOM*

Description:

A condom

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

distributeCondoms: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CORONARY_CARE_UNIT_SUPPLIES*

Description:

Supplies used in a coronary care unit

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *DELIVERY_ROOM_SUPPLIES*

Description:

Supplies used in a delivery room

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *DIET_EDUCATION_HANDOUT*

Description:

Handouts for diet education.

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

giveDietEducation: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class ELISA_SUPPLIES

Description:

Supplies used in performing ELISA

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

performHivAntiBodyTest: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class INPATIENT_SUPPLIES

Description:

Supplies used in caring for an inpatient.

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

treatCesarean: <RESULT TYPE>

Purpose:

To be specified

treatCMV: <RESULT TYPE>

Purpose:
To be specified

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:
To be specified

treatNormalTermFetus: <RESULT TYPE>

Purpose:
To be specified

treatPCP: <RESULT TYPE>

Purpose:
To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *INTENSIVE_CARE_UNIT_SUPPLIES*

Description:

Supplies used in an intensive care unit
Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class

MATERNAL_ANTI_TOBACCO_EDUCATION_HANDOUT

Description:

Handouts for maternal anti-tobacco education
Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

giveMaternalAntiTobaccoEducation: <*RESULT TYPE*>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *MATERNAL_DRUG_EDUCATION_HANDOUT*

Description:

Handouts for maternal drug education

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

giveMaternalDrugEducation: <*RESULT TYPE*>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *MATERNAL_HIV_EDUCATION_HANDOUT*

Description:

Handouts for maternal HIV education.

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

giveMaternalHivEducation: <*RESULT TYPE*>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *NEEDLE*

Description:

A needle for clean needle distribution.

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

distributeCleanNeedles: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **NEONATAL_INTENSIVE_CARE_UNIT_SUPPLIES**

Description:

Supplies used in a neonatal intensive care unit.

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

treatLowBirthWeightFetus: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class **OPERATING_ROOM_SUPPLIES**

Description:

Supplies used in an operating room

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

treatCesarean: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PCR_SUPPLIES*

Description:

Supplies used in a PCR test.

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

performHivPerTest: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PRENATAL_EDUCATION_HANDOUT*

Description:

Handouts for prenatal education.

Specializes:
Aggregates:
Operations:

givePrenatalEducation: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *SEX_EDUCATION_HANDOUT*

Description:

Handouts for sex education.

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

giveSexEducation: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *SPONGE*

Description:

The general, abstract sponge. (!)

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What specific features does a sponge have?

Class *TELEMETRY_UNIT_SUPPLIES*

Description:

Supplies used in a telemetry unit.

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *THROMBOLYTIC_SUPPLIES*

Description:

Supplies used in administering thrombolytics.

Specializes:

SUPPLY

The general supply class.

Aggregates:

Operations:

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *ULTRASOUND_SUPPLIES*

Description:

Supplies used in performing ultrasound measurements

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

measurePelvisSizeUltrasound: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *URINE_SAMPLE_SUPPLIES*

Description:

Supplies used for taking a urine sample

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

measureProteinLevelInUrine: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *WESTERN_BLOT_SUPPLIES*

Description:

Supplies used in performing a western blot test.

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

performHivAntiBodyTest: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class X-RAY_CONTRAST_AGENT_SUPPLIES

Description:

X-ray contrast agents

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

performAngiogram: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class X-RAY_SUPPLIES

Description:

Supplies used for taking x-rays

Specializes:

SUPPLY

The general supply class.

Aggregates:
Operations:

measurePelvisSizeWithXRay: <RESULT TYPE>

Purpose:
To be specified

performAngiogram: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

DRUG CLUSTER

Design issues

Intervention specific operations

As currently specified, each drug know what interventions it participates in because it has a specific operation for each intervention. There are several other possible designs: (i) drugs do not know about interventions, they have one or more operations that effect various parts of the patient; (ii) they have only one operation, they give themselves to the patient; (iii) they have know operations, the patient takes them and the implementation of the patient decides what effect they have. Approach (iii) seems the most natural, but further development will decide the issue. For the moment, I've left the intervention specific operations in the specification, if only to serve as a cross reference.

Abstract classes

Class *DRUG*

Description:

The general, abstract drug. A supply with special controls and reporting features.

Specializes:

SUPPLY

The general supply

What are the specific differentiating features of this class?
Immutable representations

Aggregates:
Operations:
Invariants:
Comments:

Differentiating features.

What are the control and reporting features that differentiate DRUG from SUPPLY? What are the features that differentiate the various specific drugs?

Mechanism of drug action

Presumably, drug objects do things to person objects. We need to define this mechanism. The various specialized drugs will presumably be different representations for this mechanism.

Mutable representations

Class *AIDS_DRUG*

Description:

<description of purpose and defining characteristics of the class>

Specializes:
Aggregates:
Operations:
Invariants:
Comments:

Specific drug

Do we have any specific AIDS drug in mind here?

Differentiating features

What are the differentiating features of this drug?

Class *ANTI_HYPERTENSIVE_DRUG*

Description:
A drug that reduces blood pressure.
Specializes:

DRUG

The general drug

Aggregates:
Operations:

treatEclampsia: <RESULT TYPE>

Purpose:
To be specified

treatPreeclampsia: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *ANTIBIOTIC*

Description:
<description of purpose and defining characteristics of the class>
Specializes:
Aggregates:
Operations:

treatPCP: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the differentiating features of this drug?

Specific drug

Do we have any specific antibiotic in mind here?

Class *ASPIRIN*

Description:

Common aspirin.

Specializes:

DRUG

The general drug

Aggregates:

Operations:

treatWithAspirin: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *AZT*

Description:

The AIDS drug AZT.

Specializes:

DRUG

The general drug

Aggregates:

Operations:

treatCMV: <RESULT TYPE>

Purpose:

To be specified

treatMaternalHIV: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *BETA_BLOCKER*

Description:

A beta blocker drug, used for treating angina.

Specializes:

DRUG

The general drug

Aggregates:
Operations:

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CALCIUM_CHANNEL_MEDICATION*

Description:
A drug used for treating angina.
Specializes:

DRUG

The general drug
Aggregates:
Operations:

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *CARDIAC_BYPASS_SURGERY_DRUG*

Description:
A drug used during cardiac bypass surgery.
Specializes:

DRUG

The general drug

Aggregates:
Operations:

performBypassSurgery: <RESULT TYPE>

Purpose:
To be specified

Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Specific drug or category?

Is this a specific drug or an abstract class with several specializations?

Class *CESAREAN_SECTION_DRUG*

Description:

A drug used in cesarean section operations..

Specializes:

DRUG

The general drug

Aggregates:

Operations:

treatCesarean: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Specific drug or category?

Is this a specific drug or an abstract class with several specializations?

Class *GANCOCLOVIR*

Description:

A drug used to treat AIDS.

Specializes:

DRUG

The general drug

Aggregates:

Operations:

treatCMV: <RESULT TYPE>

Purpose:

To be specified

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *LOVASTATIN*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

Aggregates:

Operations:

treatWithLovastatin: <RESULT TYPE>

Purpose:

To be specified

Invariants:
Comments:

Differentiating features

What are the differentiating features of this drug?
Class *Niacin*
Description:
<description of purpose and defining characteristics of the class>
Specializes:
Aggregates:
Operations:

treatWithNiacin: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the differentiating features of this drug?
Class *NITRATE*
Description:
A drug used to treat angina.
Specializes:

DRUG

The general drug

Aggregates:
Operations:

treatAnginaPainWithMedication: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?
Class *STREPTOKINASE*
Description:
<description of purpose and defining characteristics of the class>
Specializes:
Aggregates:
Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:
To be specified

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:
To be specified

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:
To be specified
Invariants:
Comments:

Differentiating features

What are the differentiating features of this drug?

Class *t-PA*

Description:

<description of purpose and defining characteristics of the class>

Specializes:

Aggregates:

Operations:

treatMIWithCCU: <RESULT TYPE>

Purpose:
To be specified

UTILITY CLUSTER

Abstract classes

Class *NAME*

Description:

<description of purpose and defining characteristics of the class>

treatMIWithTelemetryUnit: <RESULT TYPE>

Purpose:
To be specified

treatMIWithThrombolytics: <RESULT TYPE>

Purpose:
To be specified

Invariants:

Comments:

Differentiating features

What are the differentiating features of this drug?

Immutable representations

Specializes:

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What specific differentiating features does this class have?

Class ADDRESS (<ABBREVIATION>)Description:

<description of purpose and defining characteristics of the class>

Specializes:Aggregates:Operations:Invariants:Comments:**Differentiating features**

What specific differentiating features does this class have?

Class MONEYDescription:

<description of purpose and defining characteristics of the class>

Specializes:Aggregates:Operations:Invariants:Comments:**Differentiating features**

What specific differentiating features does this class have?

Class CALENDARDescription:

The general calendar

Specializes:

None

Aggregates:Operations:Invariants:Comments:**Can we buy this?**

One would think that a calendar would be the type of commonly used component one could buy. I don't think we are too particular about its features.

Class SCHEDULEDescription:

A schedule of operating hours for a resource

Specializes:**CALENDAR**

The general calendar.

Aggregates:**lengthOfDay(d: DATE): TIME**

The number of hours the resource is available on date d.

Operations:**lengthOfDay(d: DATE): TIME**Purpose:

The length of the operational day.

Require:

<preconditions of the operation>

Algorithm:
deferred

Ensure:
result > 0

open(d: DATE): TIME

Purpose:

The time the resource becomes available for use on date d.

Require:

<preconditions of the operation>

Algorithm:
deferred

Ensure:

<postconditions for the operation>

setOpen(day:DATE, o: TIME): VOID

Purpose:

Sets the open time for date *day*.

Require:

Algorithm:
deferred

Ensure:

open(d) = o and open(d) < close(d)

close(d: DATE): TIME

Purpose:

The time the resource becomes unavailable for use on date d.

Require:

<preconditions of the operation>

Algorithm:
deferred

Ensure:

<postconditions for the operation>

setClose(day:DATE, c: TIME): VOID

Purpose:

Sets the open time for date *day*.

Require:

Algorithm:
deferred

Ensure:

close(d) = c and open(d) < close(d)

Invariants:

lengthOfDay positive

for all d open(d) <= close(d)

Comments:

Enhancements

This specification is probably too simple for what we need. For instance, it allows only a single period of operation. What features do we need to add for completeness and convenience?

Class *TIME*

Description:

Some type for representing time (e.g. day time, simulation time)

Specializes:

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What specific differentiating features does this class have?

Class *DATE*

Description:

Some type for representing dates.

Specializes:

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What specific differentiating features does this class have?

Class *QUEUE[T]*

Description:

A queue of objects of type T

Specializes:

Aggregates:

storage: BOUNDED_SET

The storage in which items in the queue are placed; possibly shared with other queues.

isEmpty: BOOLEAN

True if no items in queue

isFull: BOOLEAN

True if there is no more room in storage; i.e. if *storage.isFull*

Operations:

insert(item: T)

Purpose:
Inserts *item* into the queue.

Require:
item not void and not *isFull*

Algorithm:
deferred

Ensure:
not *isEmpty* and *back* = *item*

remove: T

Purpose:
Removes the item at the front of the queue

Require:
not *isEmpty*

Algorithm:
deferred

Ensure:
result not void and *result* = old *front* and not *isFull*

back: T

Purpose:
Returns the item at the back of the queue, but does not remove it from the queue

Require:
not *isEmpty*

Algorithm:
deferred

Ensure:
result not void

front: T

Purpose:
Returns the item at the front of the queue, but does not remove it from the queue

Require:
not *isEmpty*

Algorithm:
deferred

Ensure:
result not void

Invariants:

Comments:

Shared storage

Multiple queues may share the same storage. For instance, a waiting room may be the storage for several queues for different departments or services. How do we implement this?

Answer: *storage* is a reference (pointer) to a bounded set. Several queues may be given a reference to the same storage.

Class *SET[T]*Description:

The general abstract collection of objects of type T

Specializes:Aggregates:

ct: **INTEGER**

Count; the number of items in the set

isEmpty: **BOOLEAN**

True if *ct* = 0

Operations:

insert(item: T)

Purpose:

Insert the item in the set.

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

remove(item: T)

Purpose:

Remove the item from the set.

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

contains(item: T): **BOOLEAN**

Purpose:

True if item is a member of the set.

Require:

<preconditions of the operation>

Algorithm:

<pseudocode for operation>

Ensure:

<postconditions for the operation>

Invariants:Comments:**Loose usage**

The notion of set is used somewhat loosely in this specification for any arbitrary collection. In particular, I've not worried much about whether a given item can appear in a "set" more than once. Depending on context multiset or bag might be more rigorously correct, but I've not sorted that out.

Class *BOUNDED_SET[T]*

Description:

A set of items of type T with a fixed, finite maximum number of elements

Specializes:

SET[T]

The general set.

Aggregates:

ub: INTEGER

The upper bound on the number of items in the set; the capacity of the set.

RECORDS CLUSTER

Abstract Classes

Class *RECORD*

Description:

A physical entity used to store information.

Specializes:

Aggregates:

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

isFull: BOOLEAN

True if *ct* = *ub*

Operations:

Invariants:

Bounded

ct <= *ub*

Comments:

Mutable representations

Immutable representations

Class *EKG_RECORD*

Description:

The results of an EKG measurement

Specializes:

RECORD

The general record.

Aggregates:

patient: PERSON

The patient the chart belongs to.

Operations:
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *BLOOD_PRESSURE_RECORD*

Description:

The result of a blood pressure measurement
Specializes:

RECORD

The general record

Aggregates:

systolicPressure: REAL

The systolic blood pressure reading

diastolicPressure: REAL

The diastolic blood pressure reading

Operations:
Invariants:
Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *PATIENT_CHART*

Description:

The collection of medical records maintained for a patient.
Specializes:

SET[RECORD]

The collection of records

Aggregates:
Operations:
Invariants:
Comments:

Patient reference

Patient is a direct reference to the associated patient object.
It may be more correct to have one or more patient
identifiers.

Other features

We probably need additional features here, a patient chart
must be more than just a set of records. We can add them as
they become apparent.

Mutable Representations
Immutable Representations

ACCOUNTS CLUSTER

Design Issues

Accounting model.

We've adopted a simplified model of accounting in this specification, based on the detailed analogy between accounting systems and plumbing systems (really!) develop by Purcell [iv]. Accounts store value, like tanks store water in a plumbing system. Transactions are flows of value from source accounts to destination accounts. By convention, positive flows move positive value from source to destination. A transaction thus decrements (subtracts signed value) from its source accounts and increments (adds signed value) to its destination accounts. Conservation of value requires that the sum of the source amounts in a transaction equals the sum of destination amounts.

No distinction is made between asset and liability accounts, or between income and expense accounts. Accounts can have either a positive or a negative balance and the liability and expense accounts just have negative balances under normal conditions. There is however a distinction between asset/liability accounts and income/expense accounts - the income/expense account balances are reset to zero at the beginning of each accounting period, while the asset/liability account balances are carried forward. The asset/liability accounts are thus cumulative accounts that represent the value stored in the organization which the accounting system models, while the income/expense accounts are rate accounts that represent value flows into or out of the organization, averaged over the accounting period. Net worth is the simple sum over all cumulative accounts, rather than the difference of the sum of the asset accounts and the sum of the liability accounts. Similarly, net income is the simple sum of the rate accounts,

rather than the difference of the sum of the income accounts and the sum of the expense accounts.

Abstract Classes

Class *BALANCE*

Description:

The general, abstract financial account; something with a balance, measured in *MONEY*.

Specializes:

None.

Aggregates:

isRate: BOOL

True if balance is zeroed at beginning of each accounting period.

Operations:

balance: MONEY

Purpose:

Returns the balance of the account

Require:

None

Algorithm:

deferred

Ensure:

Result = balance of account

zero: VOID

Purpose:

Sets the balance to zero.

Require:

None

Algorithm:

deferred

Ensure:

balance = 0.

Invariants:

Comments:

Name

The name for this class is not entirely satisfactory.

Class *ACCOUNTABLE*

Description:

The general, abstract, accountable object

Specializes:

None

Aggregates:

rateAggregate: SUMMARY_ACCOUNT

The value associated with the current object and its parts, recursively, during the current accounting period.

cumulativeAggregateCost: SUMMARY_ACCOUNT

The accumulated value associated with the current object and its parts, recursively.

rateIntrinsic: ACCOUNT

The value intrinsic to the current object, not including value associated with its parts, during the current accounting period.

cumulativeIntrinsicCost: ACCOUNT

The accumulated value intrinsic to the current object, not including value associated with its parts.

Operations:

Invariants:

Comments:

Top down versus bottom-up accounting

There are (at least) two ways to deal with accounting. The first (top-down) is that wholes calculate and accumulate their costs from the costs of their parts, recursively. The other is that whole gives their parts accounts to accrue expenses to, recursively, and the parts enter a transaction in the account each time a resource consuming action is taken. The latter mechanism is more like the problem domain, but requires modelling and coding the entire accounting system of the problem domain. In the former, each part is its own account.

Class *RESOURCE*

Description:

The general, abstract, multi-user resource. Intended to provide an abstract interface to the capacity and accounting aspects of a variety of resources, e.g. facilities, personnel, equipment, etc. Specializes:

ACCOUNTABLE

The general, abstract object with stored value

EXHAUSTABLE

The general object with finite capacity.

Aggregates:

quality: REAL

Some quantitative measure of the quality of the resource.

capacityCost: MONEY

The usage dependent cost of the resource, per unit of capacity. For instance, the cost per dose of Tylenol or the direct wage rate for personnel.

timeCost: MONEY

The usage independent cost of the resource, per unit time. For instance, the cost per month of leased equipment or the cost per year of retirement benefits for personnel.

acquisitionCost: MONEY

Any one time expenses associated with acquiring the resource. For instance, the construction cost of a facility or recruitment cost for personnel. Presumably capitalized and depreciated.

liquidationCost: MONEY

Any one time expenses (or income) associated with liquidating the resource.

usageTransaction: TRANSACTIONIN

The transaction the represents the flow of intrinsic value from the resource to its clients each time the resource is used.

currentUsers: SET[ACCOUNTABLE]

The current users of the resource, if any.

currentUsageStart: TIME

The absolute time the current usage started

currentUsageAmount: CAPACITY

The number of units of capacity the current usage involved

Operations:

startUsage: VOID

Purpose:

Log the beginning of a usage session

Require:

currentUsers.ct > 0

Algorithm:

currentUsageStart := current time

Ensure:

See algorithm

usageCost: MONEY

Purpose:

Evaluates the cost for the current usage

Require:

transaction not void

Algorithm:

result := *capacityCost***currentUsageAmount* +
*timeCost**(current time - *currentUsageStart*)

Ensure:

See algorithm

distributeCost(cost: MONEY): VOID

Purpose:

Distribute the cost to the destination entries in the transaction

Require:

None

Algorithm:

deferred

Ensure:

each destination entry in transaction gets the appropriate amount of value
not *transaction.hasBeenPosted*

stopUsage: VOID

Purpose:

Evaluate the cost of the current usage, enter it in *transaction* and post it

Require:

transaction not void

Algorithm:

distributeCost(usageCost)
transaction.post

Ensure:

usageCost is properly distributed to user accounts

Invariants:

validCapacity

capacity not void

transaction

transaction not void and not *transaction.hasBeenPosted* implies *transaction.start = currentUsageStart*

Comments:

usageCost

The specified algorithm does not distribute the acquisition and liquidation costs over the usage in any way. Is this the right way to do it?

Quality

I've specified the quality feature in the abstract, but is this really possible? Is it possible to talk about the quality of a resource without regard to what it is being used for?

Invariants and preconditions

Need to review invariants and preconditions wrt start and stop time and transaction has been posted.

Class *SINGLE_USER_RESOURCE*

Description:

A resource with at most one user at a time

Specializes:

RESOURCE

The general multi-user resource
redefine *transaction*

Aggregates:

transaction: TRANSACTION11

A single source, single destination transaction

Operations:

Invariants:

Comments:

Differentiating features

What are the specific differentiating features of this class?

Class *REUSEABLE_RESOURCE*

Description:

The general, abstract reusable resource: a resource with a schedule.

Specializes:

RESOURCE

The general resource.

Aggregates:

schedule: SCHEDULE

The schedule of the resource.

queue: QUEUE[TYPE?]

The queue of clients waiting to use the resource. See comment below.

maintenanceCost: MONEY

The cost per unit time of maintaining the resource.

Operations:

setSchedule(s: SCHEDULE): VOID

Purpose:

Sets the schedule of the resource.

Require:

not s.void

Algorithm:

deferred

Ensure:

schedule = s

Invariants:

valid schedule

not schedule.void

Comments:

Queue

In principle, every reusable resource needs a queue of clients waiting to use the resource. The queue may be an object explicitly defined in the problem domain, for instance a waiting room, or it may be an object defined only in the software implementation. As I've specified this design, every explicit problem domain object that serves as a queue will need to inherit from class QUEUE.

Type of client

What do we put in for the type of client waiting in the queue?

maintenanceCost

Should maintenanceCost be per unit time or per unit capacity?

Class EXHAUSTABLE

Description:

The general, abstract, object with finite capacity.

Specializes:

None

Aggregates:

capacity: CAPACITY

The capacity of the object.

Operations:

Invariants:

validCapacity

capacity not void

Comments:

Class CAPACITY

Description:

The general abstract quantitative measure of capacity.

Specializes:

None

Aggregates:

value: REAL

The numerical measure of capacity.

units: STRING

The units the capacity is measured in. See comment below.

Operations:

Invariants:

Comments:

Features

It's not yet clear what features we want this class to have.

The class is defined to support the notion of capacity associated with the general resource. Just how we will need to sue this is not clear. The indicated features are just a guess.

Units

I've specified units to be of type STRING. An enumerated type might be more appropriate.

Mutable Representations

Class ACCOUNTING_SYSTEM

Description:

A collection of accounts that models some organization.

Specializes:

Aggregates:

cumulativeAccounts: TREE[BALANCE]

The collection of cumulative accounts. Interior nodes are SUMMARY_ACCOUNTs and leaves are ACCOUNTs

rateAccounts: TREE[BALANCE]

The collection of rate accounts. Interior nodes are SUMMARY_ACCOUNTs and leaves are ACCOUNTs

beginningOfPeriod: DATE

The beginning of the current accounting period

endOfPeriod: DATE

The beginning of the current accounting period

Operations:

startAccountingPeriod: VOID

Purpose:

Starts the accounting period. Zeros or carries forward balances on accounts as appropriate.

Require:

None

Algorithm:

rateAccounts.root.reset

Ensure:

for each account in *rateAccounts* account.*isRate*
implies account.*balance* = 0

Invariants:

interior nodes are summary accounts

for each *balance* in *chartOfAccounts*,

balance.childCt != 0 implies

balance.conformsTo(*SUMMARY_ACCOUNT*) and

balance.childCt = 0 implies *balance*

conformsTo(*ACCOUNT*)

Comments:

Other operations yet to be specified

There's obviously a number of other operations. For instance, insert account into chart, set accounting period, etc.

Class *SUMMARY_ACCOUNT*

Description:

An account which aggregates other accounts. The balance of a summary account is the sum of its subaccount's balances and no direct increments or decrements are allowed.

Specializes:

BALANCE

The general, abstract account.

Aggregates:

subaccounts: SET[ACCOUNT]

The accounts which are summarized by Current.

Operations:

balance: MONEY

Purpose:

Returns the balance of the account; i.e. the sum of the balances of the subaccounts

Require:

None

Algorithm:

result = 0;

for each account in subaccounts, result +=
account.balance

Ensure:

Result = sum of subaccount balances

zero: VOID

Purpose:

Sets the balances of all subaccounts to zero.

Require:

None

Algorithm:

for each account in *subaccounts*, *account.zero*

Ensure:

All subaccounts have zero balane

insertSubAccount(a: ACCOUNT): VOID

Purpose:

Inserts *a* into *subaccounts*

Require:

a.isCumulative = isCumulative

Algorithm:

subaccounts.put(a)

Ensure:

subaccounts.contains(a)

deleteSubAccount(a: ACCOUNT): VOID

Purpose:

Deletes *a* from *subaccounts*

Require:

subaccounts.contains(a)

Algorithm:

subaccounts.delete(a)

Ensure:

not *subaccounts.contains(a)*

Invariants:

uniform type

for each *account* in *subaccounts*, *isCumulative = account.isCumulative*

Comments:

Class *ACCOUNT*

Description:

The common, mutable account.

Specializes:

BALANCE

The general account.

Aggregates:

Operations:

increment(amount: MONEY): VOID

Purpose:

Adds *amount* to *balance*.

Require:

amount ≥ 0

Algorithm:

balance += amount

Ensure:

new balance = old balance + amount

decrement(amount: MONEY): VOID

Purpose:

Subtracts *amount* from *balance*.

Require:

amount ≥ 0

Algorithm:

balance -= *amount*

Ensure:

new balance = *old balance* - *amount*

Invariants:

Comments:

Class TRANSACTION

Description:

An accounting transaction.

Specializes:

Aggregates:

hasBeenPosted: **BOOL**

True if transaction has been posted to the source and destination accounts.

sourceEntries: **SET[ACCOUNT_ENTRY]**

The source account entries for the transaction; the accounts and amounts which value flows out of.

destinationEntries: **SET[ACCOUNT_ENTRY]**

The destination account entries for the transaction; the accounts and amounts which value flows in to.

start: **TIME**

The absolute time the transaction began

end: **TIME**

The absolute time the transaction ended

doSave: **BOOL**

if true, write transaction to file after posting it.

Operations:

post: **VOID**

Purpose:

Distributes the transaction into the source and destination accounts

Require:

not *hasBeenPosted*

Algorithm:

for each *entry* in *sourceEntries*,
entry.account.decrement(entry.amount)
for each *entry* in *destinationEntries*,
entry.account.increment(entry.amount)
if *doSave* then write transaction to file

Ensure:

hasBeenPosted

Invariants:

conservation of value

sum of source entry amounts = sum of destination entry amounts

Comments:

Saving to a file

We need to specify the persistence mechanism in more detail. In particular, just what file do we save it to?

Class *ACCOUNT_ENTRY*

Description:

An entry in an account; a source or destination in a transaction.

Specializes:

Aggregates:

account: ACCOUNT

The account for the entry

amount: MONEY

The amount of the entry.

Operations:

setAccount(a: ACCOUNT): VOID

Purpose:

set *account* = a

Require:

a not void

Algorithm:

account := *a*

Ensure:

account = a

setAmount(a: MONEY): VOID

Purpose:

set *amount* = a

Require:

a not void

Algorithm:

amount := *a*

Ensure:

amount = a

Invariants:

account exists

account not void

Comments:

Class *TRANSACTION11*

Description:

A transaction with 1 source account and 1 destination account

Specializes:

TRANSACTION

The general transaction

Aggregates:

Operations:

make(src, dst: account, amount: money): VOID

Purpose:

Constructor.

Require:
src not void, dst not void, money not void

Algorithm:
sourceEntries.insert(srcEntry.make(src, amount))
destinationEntries.insert(dstEntry.make(dst, amount))

Ensure:

None

source: ACCOUNT

Purpose:
returns source account

Require:

None

Algorithm:

result := sourceEntries.item(1).account

Ensure:

result = source account

setSource(a: ACCOUNT): VOID

Purpose:

Sets the source account to *a*

Require:

a not void

Algorithm:

sourceEntries.item(1).setAccount(*a*);

Ensure:

desintation account is a

destination: ACCOUNT

Purpose:

returns destination account

Require:

None

Algorithm:

result := destinationEntries.item(1).account

Ensure:

result = destination account

setDestination(a: ACCOUNT): VOID

Purpose:

Sets the destination account to *a*

Require:

a not void

Algorithm:

destinationEntries.item(1).setAccount(*a*);

Ensure:

desintation account is a

amount: MONEY

Purpose:

The amount of the transaction

Require:

None

Algorithm:

result := *sourceEntries.item(1).amount*

Ensure:

result := *sourceEntries.item(1).amount*

setAmount(a: MONEY): VOID

Purpose:

Sets the amount of the transaction, but leaves the source and destination unchanged

Require:

a not void

Algorithm:

sourceEntries.item(1).setAmount(a)

destinationEntries.item(1).setAmount(a)

Ensure:

source and destination unchanged; both amounts = *a*

setAmountAndPost (a: MONEY): VOID

Purpose:

Sets the amount of the transaction then posts the transaction

Require:

a not void

Algorithm:

setAmount(a)

post

Ensure:

source unchanged; both amounts = *a*

Invariants:

one source, one destination

sourceEntries.ct = 1 and *destinationEntries.ct* = 1

Comments:

Class *TRANSACTION* IN

Description:

A transaction with 1 source account and a variable number (*n*) of destination accounts

Specializes:

TRANSACTION

The general transaction

Rename *amount* to *sourceAmount*

Aggregates:

Operations:

make(src: ACCOUNT, amount: money): VOID

Purpose:

Constructor.

Require:

src not void, *dst* not void, money not void

Algorithm:

sourceEntries.insert(srcEntry.make(src, amount))

Ensure:

None

source: ACCOUNT

Purpose:
returns source account

Require:
None

Algorithm:
result := sourceEntries.item(1).account

Ensure:
result = source account

setSource(a: ACCOUNT): VOID

Purpose:
Sets the source account to *a*

Require:
a not void

Algorithm:
sourceEntries.item(1).setAccount(a);

Ensure:
desintation account is a

sourceAmount: MONEY

Purpose:
The amount of the transaction

Require:
None

Algorithm:

result := sourceEntries.item(1).amount

Ensure:

result := sourceEntries.item(1).amount

setSourceAmount(a: MONEY): VOID

Purpose:

Sets the amount of the transaction, but leaves the source and destination unchanged

Require:

a not void

Algorithm:

sourceEntries.item(1).setAmount(a)

Ensure:

source and desintation unchanged; both amounts = *a*

Invariants:

one source

sourceEntries.ct = 1

Comments:

Immutable Representations