Chapter 12
Knowledge-Based UML Dynamic Models Generation from Enterprise Model in Hospital Information Management Process Example

Ilona Veitaite and Audrius Lopata

Abstract The main purpose of this paper is to present knowledge-based Enterprise model (EM) sufficiency as data repository for Unified Modelling Language (UML) models generation. UML models are one of the most usable modelling languages in system lifecycle design stage, despite the problem domain of the system. UML models can be generated from Enterprise Model by using particular transformation algorithms presented in previous researches. Generation process from Enterprise model is represented by certain Hospital Information Management process example. Generated UML dynamic Use Case, Activity, Sequence and State models of different perspectives of Hospital Information Management process prove sufficiency of stored information in Enterprise model.

Keywords UML · Enterprise model · Transformation algorithm · Knowledge-based IS engineering · Hospital IS management

12.1 Introduction

Despite the progress of all information technologies, information system (IS) engineering process still challenges professionals of this field: analysts, designers, researchers and etc. Enterprise modelling makes giant impact to successful information system design process. There are many Enterprise models and Enterprise modelling methodologies, which are applied in different ways and various types of models are built based on chosen Enterprise model [1, 2].

I. Veitaite (✉)
Institute of Social Sciences and Applied Informatics, Vilnius University, Muitinės g. 8, 44280 Kaunas, Lithuania
e-mail: ilona.veitaite@knf.vu.lt

A. Lopata
Kaunas University of Technology, Studentų g. 50, 51368 Kaunas, Lithuania
e-mail: audrius.lopata@ktu.lt

D. Kriksciuniene and V. Sakalauskas (eds.), *Intelligent Systems for Sustainable Person-Centered Healthcare*, Intelligent Systems Reference Library 205,
https://doi.org/10.1007/978-3-030-79353-1_12

Unified Modelling Language is a highly accepted among IS analysts and designers and is commonly used for IS design. It is used as standard notation to represent designed information system from different views, it provides information in both: structural and behavioural perspectives. Correctly created UML models of any problem domain can be the background for code generation and ensure the success of final IS version [1, 3–5].

Enterprise model can be used as the background for UML models. Correct UML models can be created only then, when gathered into Enterprise Model data is verified, validated and have enough quality. Data gathering process should be done under analysts and experts supervision. Enterprise model with verified and validated data of particular problem domain fully serves all necessary data. Using this data UML models can be generated from Enterprise model through transformation algorithms and after generation process these models are main source for further IS development life cycle stage [3, 4, 6–9].

12.2 Structure of Knowledge-Based Enterprise Model

EMM is formally defined EM structure, which consists of a formalized EM in line with the general principles of control theory. EM is the main source of the necessary knowledge of the particular business domain for IS engineering and IS re-engineering processes (Fig. 12.1) [6, 10].

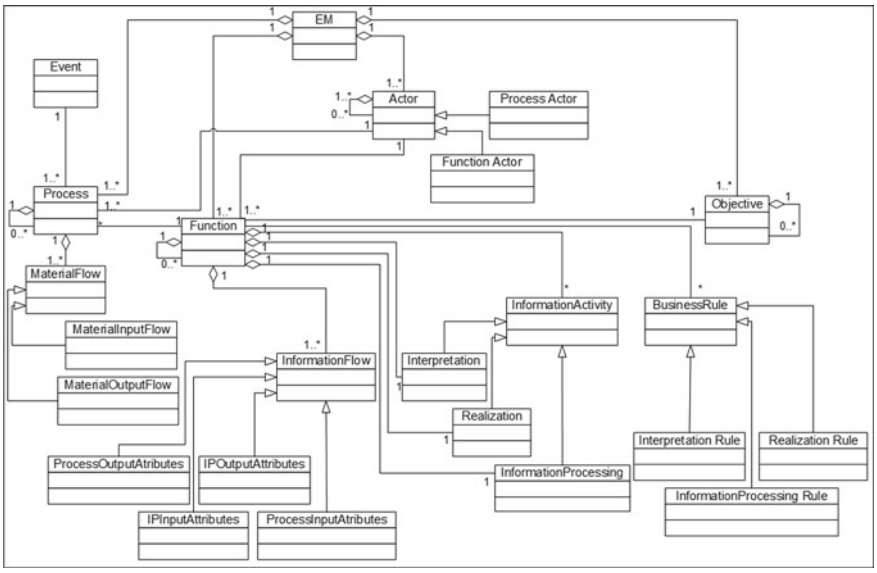

Fig. 12.1 Enterprise meta-model class diagram [6, 9, 10]

EM class model has twenty-three classes. Essential classes are Process, Function and Actor. Class Process, Function, Actor and Objective can have an internal hierarchical structure. These relationships is presented as aggregation relationship. Class Process is linked with the class MaterialFlow as aggregation relationship. Class MaterialFlow is linked with the classes MaterialInputFlow and MaterialOutputFlow as generalization relationship. Class Process is linked with Classes Function, Actor and Event as association relationship. Class Function is linked with classes InformationFlow, InformationActivity, Interpretation, InformationProcessing and Realization as aggregation relationship. These relationships define the internal composition of the Class Function. Class InformationFlow is linked with ProcessOutputAtributes, ProcessInputAtributes, IPInputAttributes and IPOutputAttributs as generalization relationship. Class InformationActivity is linked with Interpretation, InformationProcessing and Realization as generalization relationship. Class Function linked with classes Actor, Objective and BusinessRule as association relationship. Class BusinessRule is linked with Interpretation Rule, Realization Rule, InformationProcessing Rule as generalization relationship. Class Actor is linked with Function Actor and Process Actor as generalization relationship [3, 6, 10, 11].

12.3 Transformation Algorithms of UML Models from Enterprise Model

Each of structural or behavioural UML models can be generated through transformation algorithm and each of models has separate transformation algorithm. These transformation algorithms are presented in previous researches. Main focus of researches is dedicated for generation behavioural or dynamic UML models, because they are more complex and variable [9, 11–13]. To have better understanding of transformation algorithm itself, top level transformation algorithm of UML models generation from EM process is presented in the figure (Fig. 12.2) and described step by step [9, 11–13].

- Step 1: Particular UML model for generation from EM process is identified and selected.
- Step 2: If the particular UML model for generation from EM process is selected then algorithm process is continued, else the particular UML model for generation from EM process must be selected.
- Step 3: First element from EM is selected for UML model, identified previously, generation process.
- Step 4: If the selected EM element is initial UML model element, then initial element is generated, else the other EM element must be selected (the selected element must be initial element).
- Step 5: The element related to the initial element is selected from Enterprise model.

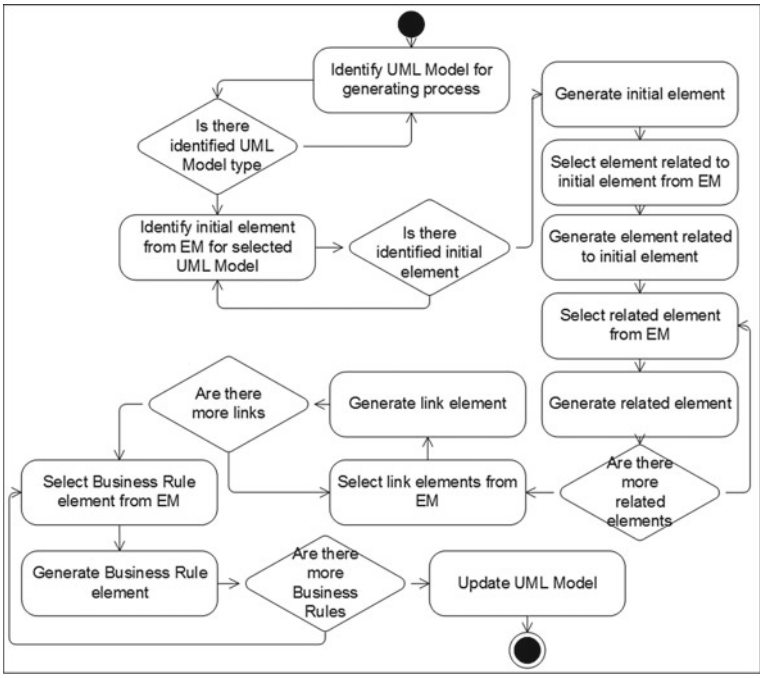

Fig. 12.2 The top level transformation algorithm of UML models generation from EM process [9, 11–13]

- Step 6: The element related to the initial element is generated as UML model element.
- Step 7: The element related to the previous element is selected from Enterprise model.
- Step 8: The element related to the previous element is generated as UML model element.
- Step 9: If there are more related elements, then they are selected from EM and generated as UML model elements one by one, else the link element is selected from Enterprise model.
- Step 10: The link element is generated as UML model element.
- Step 11: If there are more links, then they are selected from EM and generated as UML model elements one by one, else the Business Rule element is selected from Enterprise model.
- Step 12: The Business Rule element is generated as UML model element.
- Step 13: If there are more Business Rules, then they are selected from EM and generated as UML model elements one by one, else the generated UML model is updated with all elements, links and constraints.
- Step 14: Generation process is finished.

Table 12.1 Description of knowledge stored in Enterprise model

Enterprise model element	Description
Actor	In actor element can be stored information related with process or function executor. Actor element is responsible of information related with the process or function participant, it can be person, group of persons, subject such as an IS, subsystem, module and etc
Process, function	In process or function elements can be stored all information related with any user, entity, object, subject and its behaviour. Process or function element is responsible of information related with any operation, activity, status change, movement which is implemented by any actor, entity, participant and etc
Information flow	In Information Flow element can be stored diverse information flow types, such as Information input and output attributes or/and process input and output attributes. Information Flow element is responsible of information related with each element input and output attributes, details which make impact on other elements, their state or status
Business rule	In Business Rule element can be stored different rules such as interpretation, realization or/and information processing. Business rule element is responsible of information about how different elements in IS design phase are related; what restrictions and restraints are applied to these elements

Table 12.1 presents part of Enterprise model elements and their descriptions in order to describe elements, which are necessary in this particular research.

12.4 Generated UML Models of Hospital Information Management Process Example

The Hospital intended to manage outside patients is the object of presented example. In this institution a doctor is only associated with one specialized hospital department (cardiology, pediatrics, etc.) at a time. Each doctor has a visiting time and day in a week.

At reception the patient data is entered and the necessary fees are also taken. The patient is tracked on the basis of the ID number which is generated automatically.

Usually a patient can visit the doctors in two possible ways: directly selecting a doctor or by getting admitted to the hospital.

A doctor can prescribe tests based on the patient's described condition. The patient visits the laboratory to get done the tests prescribed by the doctor. The reports of the tests are given to the patient. The payments related to the tests are done at the reception. According the reports, the doctor prescribes the patient medicines or further tests, if they are needed or is asked to get admitted in hospital.

If available a patient is admitted into a ward of a particular department as per the doctor's prescription. The number of available wards is limited and if there is no free ward the admission of the patient is rescheduled.

Also in case of the prescription of the doctor the patient is operated on a scheduled date and time as decided by the doctor who is responsible for the operation.

After the finishing of the treatment a patient may get discharged on an advice of his doctor and upon the full payment of all due charges at the reception. On payment of complete dues the reception generates a discharge card for the patient.

All data of particular problem domain, in this case, Hospital Information Management data is stored in Enterprise Model described previously. Stored information in Enterprise model is already verified and validated by expert and analyst, so it is ready to use for UML model generation.

12.4.1 UML Use Case Model of Hospital Information Management Process Example

A UML Use Case model is the initial form to identify and present system requirements for a new IS underdeveloped. Use cases identify the expected behaviour—what should be done, and not the exact method of how it should be done. Main advantage of use case modelling is that it assists to design a system from the end user's view. It is a powerful technique for communicating system behaviour in the user's conditions by specifying all externally visible system behaviour [3, 7, 8].

Table 12.2 presents UML Use Case model elements generated from Enterprise model of Hospital Information Management process example. In Enterprise Model all information related with actors, their functions and relationships between these functions is stored. There are four actors: Patient, Doctor, Receptionist and Laboratory Assistant; Receptionist is related with five use cases; Laboratory Assistant—with one use case; Doctor is related with three uses cases and Patient is related with seven use cases. Four use cases includes some additional use cases, six relationships in total. These elements and their relationships are presented in the next figure.

Figure 12.3 presents UML Use Case model of Hospital Information Management process example generated step by step from Enterprise Model through UML Use Case transformation algorithm.

12.4.2 UML Activity Models of Hospital Information Management Process Example

UML Activity model describes how activities are coordinated, activities dependence from the actor or previous activity. It provides a service which can be in various levels of abstraction. Usually, an event needs to be gained by some operations, particularly

Table 12.2 UML use case model elements generated from enterprise model of hospital information management process example [3, 7, 8]

Enterprise model element	UML use case model element	Hospital information management process example	Description
Actor	Actor	Patient	There are four actors, each of them is behavioural classifier which defines a role played in particular example
		Doctor	
		Receptionist	
		Laboratory assistant	
Process, function	Use case	Laboratory visit for the test	There are fourteen use cases, each use case is a type of behavioural classifier that describes a unit of functionality performed by three actors
		Test report generation	
		Payment for the test at reception	
		Registration for treatment	
		ID generation	
		Fee payment	
		Admission to ward	
		Discharging	
		Account settlement	
		Discharging card generation	
		Test prescription	
		Test report analysis	
		Prescription for medicines	
		Operation performing	
Business rule	Include	Six include elements	There are six include elements, each include is a directed relationship between two use cases which is used to demonstrate that behaviour of the included use case is inserted into the behaviour of the including use case

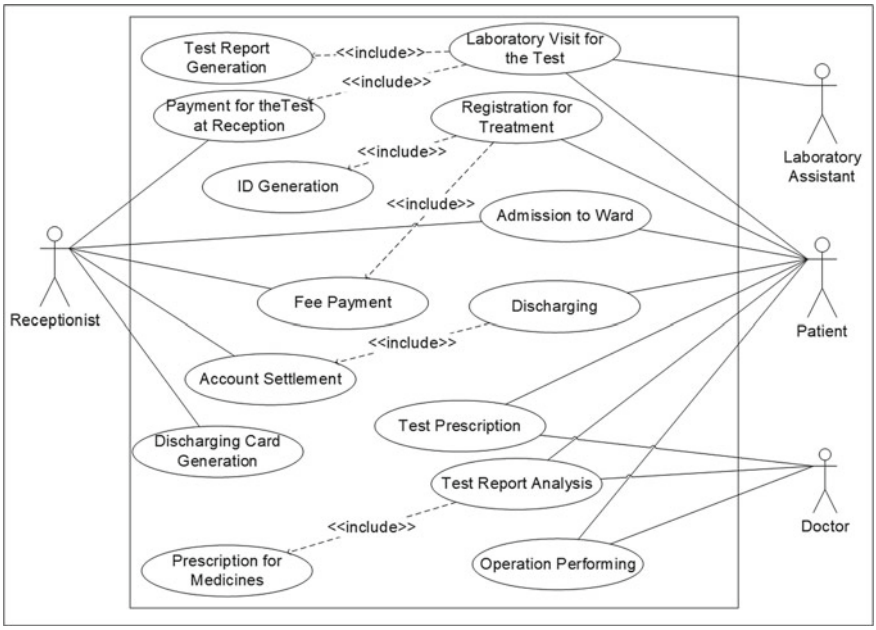

Fig. 12.3 UML use case model of hospital information management process example

where the operation is intended to gain a number of different things that require coordination, or how the events in a single use case relate to one another, especially, use cases where activities may overlap and require coordination [3, 7, 8].

According previously described UML Use Case model there is possible to identify at least five different UML Activity models: Patient Registration, Ward Assignation, Medical Tests, Treatment Process and Discharging.

UML Activity Model: Patient Registration

First UML Activity Model generated from EM is Patient Registration, where two participants—actors take part: Patient and Receptionist.

Table 12.3 presents UML Activity model elements generated from Enterprise model of Hospital Information Management process example, Registration part. Actor—first UML Activity model partition Patient starts registration process: visits reception, provides personal data, Actor—second partition Receptionist enters patient's data and provides patient's ID number, last activity Fee Payment is related with first partition, Patient pays the fee and registration process ends.

Figure 12.4 presents UML Activity Model of Hospital Information Management process example, Registration part generated step by step from Enterprise model through UML Activity model transformation algorithm [9].

Table 12.3 UML activity model elements generated from enterprise model of hospital information management process example, registration part [3, 7, 8]

Enterprise model element	UML activity model element	Hospital information management process example	Description
Actor	Partition	Patient	There are two partitions and activities are related with these actors
		Receptionist	
Function, process	Activity	Reception visit	There are five activities directly related with two partitions: patient—three activities, receptionist—two. They represent a parameterized behaviour as coordinated flow of actions
		Personal data provision	
		Data entering into system	
		Patient ID generation	
		fee payment	
Business rules	Control nodes	Initial node	There are two control nodes: one node—initial node in the beginning; final node in the end of the process
		Final node	

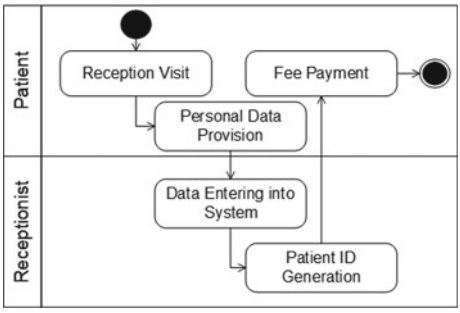

Fig. 12.4 UML activity model of hospital information management process example. registration

UML Activity Model: Ward Assignation

Second UML Activity Model generated from EM is Ward Assignation, where two participants—actors take part: Patient and Receptionist.

Table 12.4 presents UML Activity model elements generated from Enterprise model of Hospital Information Management process example, Ward Assignation

Table 12.4 UML activity model elements generated from enterprise model of hospital information management process example, ward assignation part [3, 7, 8]

Enterprise model element	UML activity model element	Hospital information management process example	Description
Actor	Partition	Patient	There are two partitions and activities are related with these actors
		Receptionist	
Function, process	Activity	Ward availability check	There are four activities directly related with two partitions: patient—one activity, receptionist—three. They represent a parameterized behaviour as coordinated flow of actions
		Provision of new dates	
		Ward assignment	
		New dates inquiry	
Information flow	Object flow edge	Ward assignation details to patient	There are two object flow edges which are activity edges used to show data flow between activities
		Ward information update to reception	
Business rules	Control nodes	Initial node	There are five control nodes: one node—initial node in the beginning; decision node—for ward availability check; join and fork nodes—to relate object flow edges, final node in the end of the process
		Decision node	
		Join node	
		Fork node	
		Final node	

part. Actor—first UML Activity model partition Receptionist starts Ward Assignation process: Checks ward availability, assigns it, or inquires for new dates, because there are no free wards, Actor—second partition Patient provides new date for ward assignation, last activities are related with first partition, Receptionist prepares information for patient and updates information in Reception and process ends.

Figure 12.5 presents UML Activity Model of Hospital Information Management process example, Ward Assignation part generated step by step from Enterprise model through UML Activity model transformation algorithm [9].

UML Activity Model: Medical Tests

Third UML Activity Model generated from EM is Medical Tests, where three participants—actors take part: Patient, Laboratory Assistant and Receptionist.

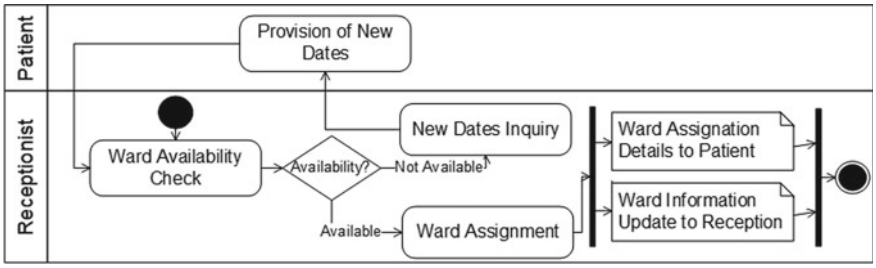

Fig. 12.5 UML activity model of hospital information management process example. Ward assignation

Table 12.5 UML activity model elements generated from enterprise model of hospital information management process example, medical tests part [3, 7, 8]

Enterprise model element	UML activity model element	Hospital information management process example	Description
Actor	Partition	Patient	There are three partitions and activities are related with these actors
		Laboratory assistant	
		receptionist	
Function, process	Activity	Laboratory visit for test	There are ten activities directly related with three partitions: patient—four activities, laboratory assistant—five activities, receptionist—one. They represent a parameterized behaviour as coordinated flow of actions
		Doctor's prescription check	
		Sample inquiry	
		Sample provision	
		Performing of the test	
		Payment order generation	
		Report generation	
		Fee payment	
		Issuing receipt	
		Payment receipt provision	
Business rules	Control nodes	Initial node	There are four control nodes: one node—initial node in the beginning; join and fork nodes—to relate additional activities; final node in the end of the process
		Join node	
		Fork node	
		Final node	

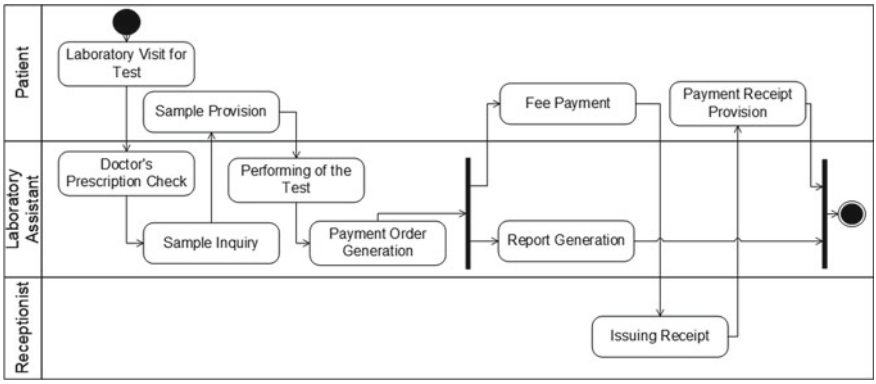

Fig. 12.6 UML activity model of hospital information management process example. Medical tests

Table 12.5 presents UML Activity model elements generated from Enterprise model of Hospital Information Management process example, Medical Tests part. Actor—first UML Activity model partition Patient starts Medical Tests process: visits laboratory and provides sample after inquiry, Actor—second partition Laboratory checks doctor's prescription, inquires for sample, performs test, generates payment order and prepares report for the doctor; Actor—third partition Receptionist confirms payment form the patient and provides receipt; Patient makes payment and after payment confirmation receives receipt and process ends.

Figure 12.6 presents UML Activity Model of Hospital Information Management process example, Medical Tests part generated step by step from Enterprise model through UML Activity model transformation algorithm [9].

UML Activity Model: Treatment Process

Fourth UML Activity Model generated from EM is Medical Tests, where two participants—actors take part: Patient and Doctor.

Table 12.6 presents UML Activity model elements generated from Enterprise model of Hospital Information Management process example, Treatment Process part. Actor—first UML Activity model partition Doctor starts Treatment Process: meets the patient, analyses provided test reports, regarding test results decides to discharge patient or continue treatment process. Doctor decides if there is need to do more tests or not, assigns treatment method medicine or operational intervention, after actor—second partition Patient confirmation, Doctor performs operation and process ends.

Figure 12.7 presents UML Activity Model of Hospital Information Management process example, Treatment Process part generated step by step from Enterprise model through UML Activity model transformation algorithm [9].

Table 12.6 UML activity model elements generated from enterprise model of hospital information management process example, treatment process part [3, 7, 8]

Enterprise model element	UML activity model element	Hospital information management process example	Description
Actor	Partition	Patient	There are two partitions and activities are related with these actors
		Doctor	
Function, process	Activity	Patient visit	There are ten activities directly related with two partitions: patient—two activities, doctor—eight. They represent a parameterized behaviour as coordinated flow of actions
		Test report provision	
		Report analysis	
		Issuing discharge	
		Test requirements check	
		Test prescription	
		Treatment requirement check	
		Operation scheduling	
		Confirmation of operation	
		Performing operation	
Business rules	Control nodes	Initial node	There are five control nodes: one node—initial node in the beginning; three decision nodes—for test report status, for more tests possibility, for treatment type; final node in the end of the process
		Decision nodes	
		Final node	

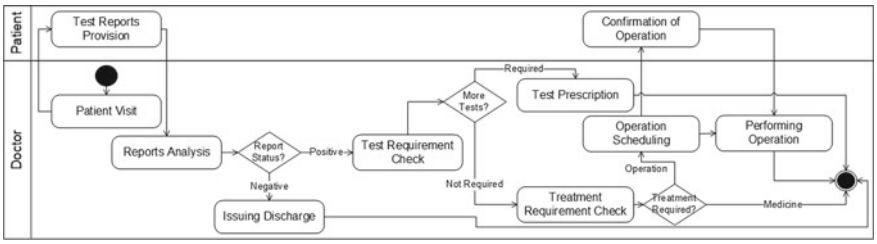

Fig. 12.7 UML activity model of hospital information management process example. Treatment process

Table 12.7 UML activity model elements generated from enterprise model of hospital information management process example, discharging part [3, 7, 8]

Enterprise model element	UML activity model element	Hospital information management process example	Description
Actor	Partition	Patient	There are two partitions and activities are related with these actors
		Receptionist	
Function, process	Activity	Approaching with discharge advice	There are six activities directly related with two partitions: patient—two activities, receptionist—four. They represent a parameterized behaviour as coordinated flow of actions
		Data check	
		Discharge card generation	
		Payment check order	
		Due amount payment	
		Discharge card provision	
Business rules	Control nodes	Initial node	There are two control nodes: one node—initial node in the beginning; decision node—for payment status final node in the end of the process
		Decision node	
		Final node	

UML Activity Model: Discharging

Fifth UML Activity Model generated from EM is Discharging, where two participants—actors take part: Patient and Receptionist.

Table 12.7 presents UML Activity model elements generated from Enterprise model of Hospital Information Management process example, Discharging part. Actor—first UML Activity model partition Patient starts discharging process: approaches with discharge advice from the doctor, Actor—second partition Receptionist checks data, generate discharge card, check payment status, after patient makes payment, provides discharge card and process ends.

Figure 12.8 presents UML Activity Model of Hospital Information Management process example, Discharging part generated step by step from Enterprise model through UML Activity model transformation algorithm [9].

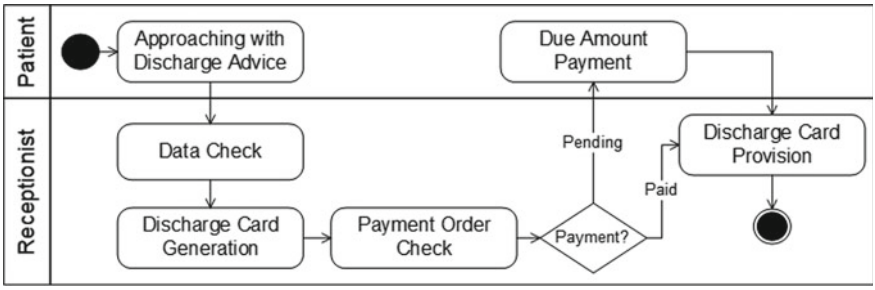

Fig. 12.8 UML activity model of hospital information management process example. Discharging

12.4.3 UML Sequence Models of Hospital Information Management Process Example

UML Sequence model is an interaction model that defines how operations are implemented. This model fixates the interaction between objects in the context of a collaboration. UML Sequence model is time focus and it shows the order of the interaction visually by using the vertical axis of the diagram to deliver time what messages are sent and when [3, 7, 8].

According previously described UML Use Case and UML Activity models there is possible to identify at least three different UML Sequence models: Patient Admission, Tests and Treatment, and Discharging.

UML Sequence Model: Patient Admission

First UML Sequence Model generated from EM is Patient Admission, where four participants—Lifelines take part: Patient, Receptionist, Database and Ward.

Table 12.8 presents UML Sequence model elements generated from Enterprise model of Hospital Information Management process example, Patient Admission part. In Enterprise Model all information related with actors and their collaboration is stored. There are four actors—process participants, which are called Lifelines in UML Sequence model: persons—Patient, Receptionist, subject—Database, object—Ward. Patient registers to the hospital, Receptionist enters gathered data, Patient requests for the ward, Receptionist checks availability and confirms or denies ward availability.

Figure 12.9 presents UML Sequence model of Hospital Information Management process example, Patient Admission part generated step by step from Enterprise model through UML Sequence model transformation algorithm [9].

UML Sequence Model: Tests and Treatment

Second UML Sequence Model generated from EM is Tests and Treatment, where four participants—Lifelines take part: Patient, Doctor, Operation and Test.

Table 12.9 presents UML Sequence model elements generated from Enterprise model of Hospital Information Management process example, Test and Treatment

Table 12.8 UML sequence model elements generated from enterprise model of hospital information management process example, patient admission [3, 7, 8]

Enterprise model element	UML sequence model element	Hospital information management process example	Description
Actor	Lifeline	Patient	There are four actors, in UML Sequence model four Lifelines, which are shown using a symbol that consists of a rectangle forming its "head" followed by a vertical line and these lines represent the lifetime of the actor—participant of the process
		Receptionist	
		Database	
		Ward	
Process, function	Message	Register(data)	There are eleven messages, related with actors and they define a communication between these actors
		Addnew(data)	
		Return	
		Return	
		Wardrequest()	
		Availabilitycheck()	
		Return(status)	
		[not available] return(n/a)	
		[if available] wardupdate(data)	
		Return	
		Return(noward)	
Business rules	Execution specification	Five execution specifications	Each of five executions specification element represents a period in the actor's lifetime

part. In EM all information related with actors-lifelines and their collaboration is stored. There are four actors—process participants, which are called Lifelines in UML Sequence model: persons—Patient, Receptionist, objects—Operation, Test. Doctor performs check-up and prescribes medicine, if necessary prescribes test, Patient provides samples and gets reports, Doctor reviews reports and prescribes more medicine or prescribes operation if necessary, prescribes more tests and operates.

Figure 12.10 presents UML Sequence model of Hospital Information Management process example, Test and Treatment part generated step by step from Enterprise model through UML Sequence model transformation algorithm [9].

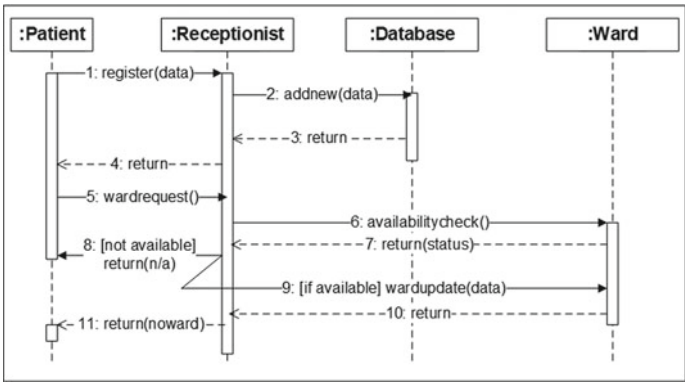

Fig. 12.9 UML sequence model of hospital information management process example, patient admission

UML Sequence Model: Discharging

Third UML Sequence Model generated from EM is Discharging, where five participants—Lifelines take part: Doctor, Patient, Reception, Database and Ward.

Table 12.10 presents UML Sequence model elements generated from Enterprise model of Hospital Information Management process example, Discharging part. In EM all information related with lifelines and their cooperation is stored. There are five actors—process participants, which are called Lifelines in UML Sequence model: persons—Patient, Doctor, subjects—Reception, Database, object—Ward. Doctor provides discharge advice, Patient requests for discharge, Reception checks information related with payments, inquires for payment, Patient makes the payment, Reception updates financial information in Database, provides Receipt, Reception updates discharge information and information related with ward, in the end Reception provides Discharge card.

Figure 12.11 presents UML Sequence model of Hospital Information Management process example, Discharging part generated step by step from Enterprise model through UML Sequence model transformation algorithm [9].

12.4.4 UML State Models of Hospital Information Management Process Example

UML State Model demonstrates the diverse states of an entity. State model can also show how an entity responds to various events by changing from one state to another [3, 7, 8].

According previously described UML models there is possible to identify at least three different UML State models describing states of: Patient, Doctor and Ward.

Table 12.9 UML sequence model elements generated from enterprise model of hospital information management process example, test and treatment [3, 7, 8]

Enterprise model element	UML sequence model element	Hospital information management process example	Description
Actor	Lifeline	Patient	There are four actors, in UML Sequence model four Lifelines, which are shown using a symbol that consists of a rectangle forming its "head" followed by a vertical line and these lines represent the lifetime of the actor—participant of the process
		Doctor	
		Operation	
		test	
Process, function	Message	Performcheckup()	There are thirteen messages, related with actors and they define a communication between these actors
		Return	
		Prescribemedicine()	
		Return	
		Prescribetest()	
		Providesamples(samples)	
		Return(report)	
		Inquirereview(reports)	
		Prescribemedicine()	
		Prescribeoperation()	
		Moretest()	
		Getoperated()	
		Operate()	
Business rules	Execution specification	Five execution specifications	Each of five executions specification element represents a period in the actor's lifetime

UML State Model: Patient

First UML State Model generated from EM describes states of Patient.

Table 12.11 presents UML State model elements generated from Enterprise model of Hospital Information Management process example, Patient part. In Enterprise Model all information related with processes, functions and their states is stored. This model is from Patient's perspective. In the model these elements are presented: initial state which starts the process, first state Patient registered, it's state changes after doctors visit: patient receives treatment, additional doctor's visit, after which doctor advices discharge procedure and patient's state changes again, patient is discharged, process ends with final state.

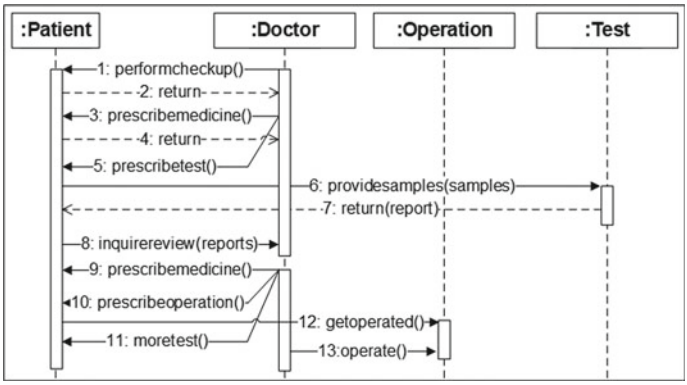

Fig. 12.10 UML sequence model of hospital information management process example, test and treatment

Figure 12.12 presents UML State model of Hospital Information Management process example, Patient part.

UML State Model: Doctor

Second UML State Model generated from EM describes states of Doctor.

Table 12.12 presents UML State model elements generated from Enterprise model of Hospital Information Management process example, Doctor part. In Enterprise Model all information related with processes, functions and their states is stored. This model is from Doctor's perspective. In the model these elements are presented: initial state which starts the process, first state Doctor registered, it's state changes after patient registers to the visit: Doctor prescribes treatment, check's up treatment results, after patient's discharge procedure Doctor's state changes, he is not needed for this particular patient, process ends with final state.

Figure 12.13 presents UML State model of Hospital Information Management process example, Doctor part.

UML State Model: Ward

Third UML State Model generated from EM describes states of Ward.

Table 12.13 presents UML State model elements generated from Enterprise model of Hospital Information Management process example, Ward part. In Enterprise Model all information related with processes, functions and their state is stored. This model is from Ward's perspective. In the model these elements are presented: initial state which starts the process, first state means ward is free, its state changes after request to occupy; after patient is discharged ward state changes again to free.

Figure 12.14 presents UML State model of Hospital Information Management process example, Ward part.

With the help of Hospital Information Management process example result of four UML models: Use Case, Activity, Sequence and State generation from Enterprise Model through transformation algorithms is presented in detailed way, all models

Table 12.10 UML sequence model elements generated from enterprise model of hospital information management process example, discharging [3, 7, 8]

Enterprise model element	UML sequence model element	Hospital information management process example	Description
Actor	Lifeline	Doctor	There are five actors, in UML Sequence model four Lifelines, which are shown using a symbol that consists of a rectangle forming its "head" followed by a vertical line and these lines represent the lifetime of the actor—participant of the process
		Patient	
		Reception	
		Database	
		Ward	
Process, function	Message	Dischargeadvice()	There are fifteen messages, related with actors and they define a communication between these actors
		Requestdiscgarge()	
		Checkdues(patientid)	
		Return(dues)	
		Askpayment(amount)	
		Paydues(amount,patientid)	
		Update(amount,patientid)	
		Return(receipt)	
		Return(receipt)	
		Return	
		Updatedischargedata(patientid)	
		Updateward()	
		Return	
		Return	
		Grantdicharge(dichargecard)	
Business rules	Execution specification	Ten Execution Specifications	Each of ten executions specification element represents a period in the actor's lifetime. Each parallels defines potentially parallel execution of behaviors of the operands of the combined fragment
	Parallel	Two parallels	

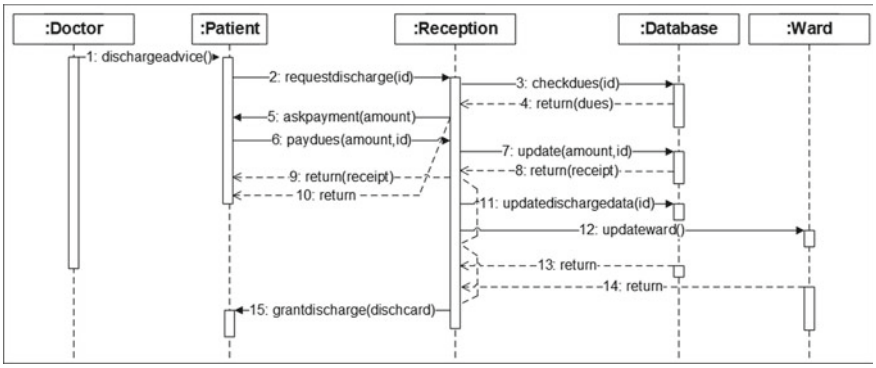

Fig. 12.11 UML sequence model of hospital information management process example, discharging

Table 12.11 UML state model elements generated from enterprise model of hospital information management process example, patient [3, 7, 8]

Enterprise model element	UML state model element	Hospital information management process example	Description
Process, function	Transition	Doctor visit	Transitions from one state to the next respond to the activities, events, what causes the state's change
		Doctor review	
		Issue to discharge	
Information flow	Simple state	Patient registered	Internal activities compartment holds a list of internal actions or state (do) activities (behaviours) that are performed while the element is in the state
		Treatment in progress	
		Discharged	
Business rules	Initial and final states	Initial state	Initial and final states are a special kind of states signifying the beginning and closing processes of defined states
		Final state	

define same example, but from different perspectives. In almost each subsection of the described example there are more than one model of the same type presented: generated UML Use Case model presents all participants (Actors), which are involved in Hospital Information Management process and their functions/processes (Use Cases); generated UML Activity models illustrate different activities from different perspectives (Registration, Ward Assignation, Medical Tests, Treatment process and Discharging) of the same example, and it is not final list of possible models of

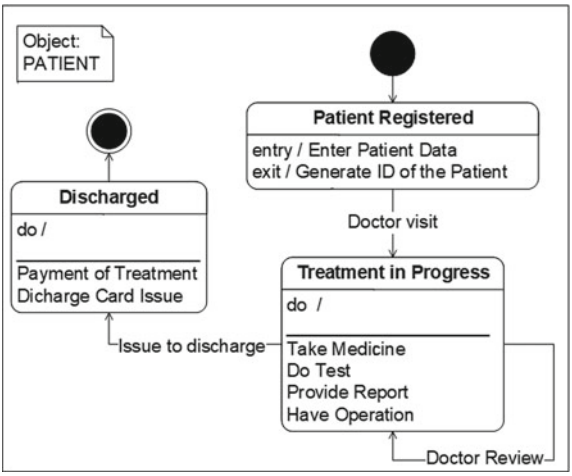

Fig. 12.12 UML state model of hospital information management process example, patient

Table 12.12 UML state model elements generated from enterprise model of hospital information management process example, doctor [3, 7, 8]

Enterprise model element	UML state model element	Hospital information management process example	Description
Process, function	Transition	Patient Registered	Transitions from one state to the next respond to the activities, events, what causes the state's change
		Patient (re)Checkup	
		Planned leave of the Doctor	
Information flow	Simple state	Doctor registered	Internal activities compartment holds a list of internal actions or state (do) activities (behaviours) that are performed while the element is in the state
		Appointing treatment	
		Doctor inactive	
Business rules	Initial and final states	Initial state	Initial and final states are a special kind of states signifying the beginning and closing processes of defined states
		Final state	

the same type; generated UML Sequence model also define sequence processes and functions sequences from different perspectives (Patient Admission, Tests and Treatment, Discharging) of the same example, and it is also not final list of possible UML Sequence models; generated UML State model describe different states from the perspectives of objects (Patient, Doctor and Ward), and states of more objects of the same example can be generated.

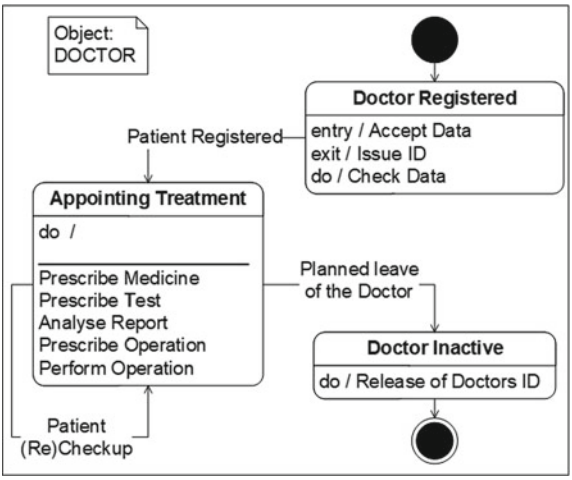

Fig. 12.13 UML state model of hospital information management process example, patient

Table 12.13 UML state model elements generated from enterprise model of hospital information management process example, Ward [3, 7, 8]

Enterprise model element	UML state model element	Hospital information management process example	Description
Process, function	Transition	Request to occupy Patient discharged	Transitions from one state to the next respond to the activities, events, what causes the state's change
Information flow	Simple state	Free Occupied	Internal activities compartment holds a list of internal actions or state (do) activities (behaviours) that are performed while the element is in the state
Business rules	Initial and final states	Initial state	Initial state is a special kind of states signifying the beginning process of defined states

Provided example of Hospital Information Management process shows and confirms, that it is not the final amount of UML models, which can be generated from EM, there are more different perspectives for UML models generation of the same example. As stated previously, knowledge-based Enterprise model which stores verified and validated data of a specific problem domain is enough data storage for generation various UML models.

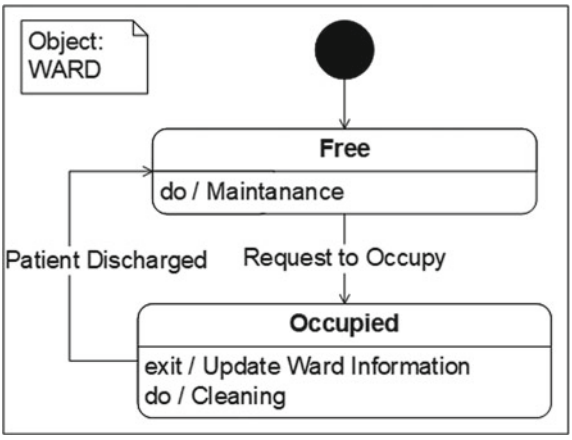

Fig. 12.14 UML state model of hospital information management process example, ward

12.5 Conclusions

The first part of the paper presents the structure of knowledge-based Enterprise model, by defining all its components and their relations.

The second part deals with the presentation of UML models generation form Enterprise model top level transformation algorithm, which is defined structurally step by step. In this section also part of Enterprise model elements, necessary for particular example of the research, are presented and their descriptions are also provided.

The third part presents Hospital Information Management process example, which data is stored in knowledge-based Enterprise model and used for UML models generation. There are defined four types of UML dynamic models in details, which represent data of chosen example in different perspectives. Each presented UML model is generated through certain transformation algorithms introduced in previous researches.

Each subsection describes different type of UML dynamic model and presents particular type of UML models variations: same UML model type, but different data used for different perspectives. All these UML models are generated based the knowledge stored in Enterprise model.

The defined Hospital Information Management process example shows that verified and validated knowledge stored in Enterprise model is sufficient for UML models generation process; that stored in Enterprise model elements are enough to transfer all UML models elements, despite the perspective of certain UML model. Using UML models generated from Enterprise Model full IS development life cycle design stage can be implemented as knowledge-based process.

Acknowledgements This publication is based upon work from COST Action "**European Network for cost containment and improved quality of health care-CostCares**" (CA15222), supported by COST (European Cooperation in Science and Technology)

COST (European Cooperation in Science and Technology) is a funding agency for research and innovation networks. Our Actions help connect research initiatives across Europe and enable scientists to grow their ideas by sharing them with their peers. This boosts their research, career and innovation.
https://www.cost.eu

References

1. Dunkel, J., Bruns, R.: Model-driven architecture for mobile applications. In: Proceedings of the 10th Inter-national Conference on Business Information Systems (BIS), Vol. 4439/2007, pp. 464–477 (2007)
2. Eichelberger, H., Eldogan, Y., Schmid, K.A.: Comprehensive Analysis of UML Tools, their Capabilities and Compliance. Software Systems Engineering. Universität Hildesheim. versio 2.0 (2011)
3. Jacobson, I., Rumbaugh, J., Booch, G.: Unified Modeling Language User Guide, 2nd edn. Addison-Wesley Professional. ISBN: 0321267974 (2005)
4. Jenney, J.: Modern methods of systems engineering: with an introduction to pattern and model based methods. ISBN-13:978-1463777357 (2010)
5. Sajja, P.S., Akerkar, R.: Knowledge-Based Systems for Development. Advanced Knowledge Based Systems: Model, Applications & Research, Vol. 1 (2010)
6. Gudas, S.: Architecture of Knowledge-Based Enterprise Management Systems: a Control View. In: Proceedings of the 13th world multiconference on systemics, cybernetics and informatics (WMSCI2009),) July 10–13, Orlando, Florida, USA, Vol. III, pp. 161–266. ISBN -10: 1-9934272-61-2 (Volume III). ISBN -13: 978-1-9934272-61-9 (2009)
7. OMG UML (2019) Unified Modeling Language version 2.5.1. Unified Modelling. https://www.omg.org/spec/UML/About-UML/
8. UML Diagrams: UML diagrams characteristic (2012). http://www.uml-diagrams.org
9. Veitaitė, I., Lopata, A.: Transformation algorithms of knowledge based UML dynamic models generation. Business information systems workshops BIS 2017, Poznan, Poland, 28–30 June. Abramowicz, W, (ed.), Lecture Notes in Business Information Processing, Vol. 303. Springer International Publishing, Cham (2017)
10. Gudas, S.: Informacijos sistemų inžinerijos teorijos pagrindai/Fundamentals of Information Systems Engineering Theory. (Lithuanian) Vilnius University. ISBN 978-609-459-075-7 (2012)
11. Veitaitė, I., Lopata, A.: Problem domain knowledge driven generation of UML models. In: Damaševičius, R., Vasiljevienė, G. (eds.), Information And Software Technologies: 24th International Conference, ICIST 2018, Vilnius, Lithuania, October 4–6, 2018: Proceedings. Springer, Cham (2018)
12. Veitaite, I., Lopata, A.: Knowledge-Based Transformation Algorithms of UML Dynamic Models Generation from Enterprise Model. In: Dzemyda, G., Bernatavičienė, J., Kacprzyk, J. (eds.), Data Science: New Issues, Challenges and Applications. Studies in Computational Intelligence, Vol. 869. Springer, Cham (2020)
13. Veitaite, I., Lopata, A.: Knowledge-Based Generation of the UML Dynamic Models from the Enterprise Model Illustrated by the Ticket Buying Process Example. In: Lopata, A., Butkienė, R., Gudonienė, D., Sukackė, V. (eds.), Information and Software Technologies. ICIST 2020. Communications in Computer and Information Science, Vol. 1283. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59506-7_3