Synthesis of two Anticipatory Models in Design and Life-Cycle of Hospitals

Ivan Krivy, Eugene Kindler
University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 03 Ostrava 1, Czech Republic
FAX: +420 596 120 478, +420 221 914 286
E-mail: ivan.krivy@osu.cz, evzen.kindler@mff.cuni.cz
http://www.osu.cz

Abstract:
A part of the human society, which either designs or manages a patient-in-bed sector of a hospital is an anticipatory system. A modern way to manage is to use (simulation) models of the sector. The exact rules holding for such systems (called patients-in-bed systems – shortly PBS) are presented in a form of axioms and for the purpose of computer models in a form of classes common for the object-oriented paradigm (OOP). The classes were applied to implement simulation models M^a oriented to the design and M^b oriented to the operation of PBS. A synthesis of both the models was made so that M^b was nested in M^a. The reason for the synthesis and the ways to overcome the difficulties are described in the paper. The two ways for to overcome the difficulties discover two ways for viewing to the anticipatory abilities of PBS.

Keywords: Health care, Patient dynamics, Anticipation in hospitals, Simulation, Nested anticipation

1 Introduction – Characteristics of the Studied Systems

Patients-in-bed systems (further PBS) of hospitals are composed of two parts, namely of a static part called ***patient-bed-fond*** (further PBF) and dynamic part containing patients. PBF is composed of rooms that contain beds. The rooms are grouped into two sections – ***standard one*** (further SC) and ***intensive care unit*** (further ICU). Time is represented in the Newtonian conception as a continuous ordered real number interval T. The patients can enter and leave beds in any moment of T but when they enter a bed they are supposed to be there at least one day.

At the beds, patients can be placed. In reality, they come into relations with other components of the hospitals – they undergo surgical interventions, X-ray studies, laboratory tests and other investigations, radiotherapy and other diagnostic and therapeutic interventions, but these relations are neglected in the "system abstraction" concerning the patients-in-bed systems, as they do not take any longer than one day. The only influence of the mentioned relations can be a transfer of a patient from SC into ICU or vice versa. The patients can be classified according to their sex and, therefore, "male rooms" and "female rooms" can be distinguished. The distinguishing is dynamic, i.e. a male room can be changed to a female room and vice versa. The mentioned

International Journal of Computing Anticipatory Systems, Volume 18, 2006
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-04-0

classification need not hold strictly – e.g. the rooms of ICU are often "mixed" and so the rooms of SC in many children PBS. In the "system abstraction", the last case, i.e. the case of absence of importance of the sex in children clinic, can be viewed so that all patients are considered to be of the same sex. In the same "system abstraction", the change of patients' sex can be neglected and such exceptional cases as Siamese twins as well, and similarly the "life rules" of patients can be limited to the following schemes:

(i) entering the standard section and then leaving the PBS,

(ii) entering the standard section, then transfer to ICU, then return to the standard section and finally leaving the PBS,

(iii) immediate entering ICU, then return to the standard section and finally leaving the PBS.

The patients of type (i) are called ***standard*** patients, those of type (ii) are called ***serious*** ones and those of type (iii) are called ***emergent*** patients.

2 Relation to the Anticipatory Systems

It is possible to state that modern health care is an anticipatory system or – better – that many view points to the modern health care offer many anticipatory systems that can be recognized at it. From the view point of PBS, two views (leading to two sorts of anticipatory systems) are important:

(a) when a PBS is being designed,

(b) when a PBS exists and is applied.

Each of these cases represents a different type of anticipatory systems. In case (a), the human society views the PBS as an object of design and in the center of the interest there is anticipation concerning years and oriented to the configuration (dimensioning and structuring) of PBF. Let such an anticipatory system be identified S^a. In case (b), the human society views the PBS as an object of control and in the center of interest there is placing the patients in rooms, while the configuration of PBF is respected as unchangeable. Let such an anticipatory system be identified S^b.

Note that case (a) is not limited to physical building of (parts of) hospitals but exists also when (a part of) an existing (building of a) hospital is reorganized, when it should be governed by certain new control rules.

In modern techniques of processes mentioned both under (a) and under (b), the systems S^a and S^b are anticipatory systems in a weak sense and can use of computer simulation models. It is evident, that the simulation model M^a applied by S^a and the simulation model M^b applied by S^b have a lot of similar properties, though they are not equal.

3 The Common Properties of Patients-in-bed Systems

The common properties of PBS can be described in the following axiomatic form, based on the first order predicate calculus.

At first, some predicates and relations should be introduced:

$R(x,t)$	At time t, x is a room	$\mu(x,t)$	At time t, x is a man
$S(x,t)$	At time t, x is a standard room	$\Phi(x,t)$	At time t, x is a female room
$B(x,t)$	At time t, x is a bed	$M(x,t)$	At time t, x is a male room
$P(x,t)$	At time t, x is a patient	$L(p,b,t)$	At time t, p occupies (bed) b
$\phi(x,t)$	At time t, x is a woman	$E(b,r,t)$	At time t, b is installed in r

The first group of axioms concerns the constant structure of PBF and of patiens‘ sex:

Axiom	Formulation of the axiom	Interpretation of the axiom
(A1)	$\forall x\, (\exists t\, P(x,t) \rightarrow \forall t\, P(x,t))$	If sometimes x is a patient, it is always patient
(A2)	$\forall x\, (\exists t\, B(x,t) \rightarrow \forall t\, B(x,t))$	If sometimes x is a bed, it is always bed
(A3)	$\forall x\, (\exists t\, R(x,t) \rightarrow \forall t\, R(x,t))$	If sometimes x is a room, it is always room
(A4)	$\forall x\, (\exists t\, \mu(x,t) \rightarrow \forall t\, \mu(x,t))$	If sometimes x is a man, it is always man
(A5)	$\forall x\, (\exists t\, \phi(x,t) \rightarrow \forall t\, \phi(x,t))$	If sometimes x is a woman, it is always woman
(A6)	$\forall x\, (\exists t\, E(x,y,t) \rightarrow \forall t\, E(x,y,t))$	If sometimes x is (a bed) in (a room) y, it is always in that room

These axioms permit simplifying of the next formulations: the time argument of the just mentioned relations can be omitted and simply $P(x)$, $B(x)$, $R(x)$, $\mu(x)$, $\phi(x)$ and $E(x,y)$ will be written. The next axioms concern the dependence among the sexes and patients:

(A7)	$\forall x\, (\mu(x) \rightarrow P(x))$	Every man is a patient
(A8)	$\forall x\, (\phi(x) \rightarrow P(x))$	Every woman is a patient
(A9)	$\forall x\, (P(x) \rightarrow \mu(x) \vee \phi(x))$	Every patient is either a man or a woman
(A10)	$\forall x \sim (\mu(x)\ \&\ \phi(x))$	No entity is both a man and a woman

A certain consequence of the axioms *(A4)*, *(A5)* and *(A10)* is commonly known from the real life:

(C1)	$\sim (\exists x\, \exists t\, \exists u\, (\mu(x,t)\ \&\ \phi(x,u)))$	Nothing can be sometimes a man and sometimes a woman

Now it is possible to discriminate among the patients, beds and rooms:

(A11)	$\forall p\, (P(p) \rightarrow \sim (B(p) \vee R(p)))$	No patient is a bed or a room
(A12)	$\forall b\, (B(b) \rightarrow \sim R(b))$	No bed is a room

There are many consequences of the introduced axioms. Let us mention the most important of them; their identifications are in the first column of the next table, while the last column points to what implies the consequences.

(C2)	$\forall b\ (B(b) \rightarrow {\sim}P(b))$	No bed is a patient	*(A11)*
(C3)	$\forall r\ (R(r) \rightarrow {\sim}(P(r) \vee B(r))$	No room is a patient or a bed	*(A11), (A12)*
(C4)	$\forall p\ (\mu(p) \rightarrow {\sim}(B(p) \vee R(p)))$	No man is a bed or a room	*(A7), (A11)*
(C5)	$\forall p\ (\phi(p) \rightarrow {\sim}(B(p) \vee R(p)))$	No woman is a bed or a room	*(A8), (A11)*
(C6)	$\forall b\ (B(b) \rightarrow {\sim}\phi(b))$	No bed is a woman	*(C2), (A8)*
(C7)	$\forall b\ (B(b) \rightarrow {\sim}\mu(b))$	No bed is a man	*(C2), (A7)*
(C8)	$\forall r\ (R(r) \rightarrow {\sim}\phi(r))$	No room is a woman	*(C3), (A8)*
(C9)	$\forall r\ (R(r) \rightarrow {\sim}\mu(r))$	No room is a man	*(C3), (A7)*

The following axioms express the basic relations among the patients and beds:

(A13)	$\forall x\ \forall y\ \forall t\ (L(x,y,t) \rightarrow P(x)\ \&\ B(y))$	The relation of occupying exists only between a patient and a bed
(A14)	$\forall p\ \forall b\ \forall t\ (P(p)\ \&\ B(b)\ \&\ L(p,b,t) \rightarrow \forall c\ (L(p,c,t) \rightarrow c=b))$	A patient cannot occupy more beds at the same time
(A15)	$\forall p\ \forall b\ \forall t\ (P(p)\ \&\ B(b)\ \&\ L(p,b,t) \rightarrow \forall q\ (L(q,b,t) \rightarrow p=q))$	A bed cannot be occupied by more patients at the same time

In real life, it is possible that a patient is not always placed at a bed (as he can be outside the PBD, i.e. outside the studied system), that a bed is not always occupied and that a patient can be stepwise placed at different beds. Those aspects can be satisfied also under the presented axioms, because the following statements are not among the consequences of the axioms:

$$\forall p\ (P(p) \rightarrow\ \forall t\ \exists b\ L(p,b,t))$$

$$\forall b\ (B(p)\ \rightarrow \forall t\ \exists p\ L(p,b,t))$$

$$\forall p\ \forall b\ \forall t\ \forall u\ (P(p)\ \&\ B(b)\ \&\ L(p,b,t) \rightarrow \forall c\ \forall u\ (L(p,c,u) \rightarrow c=b))$$

The further pair of axioms concerns relations among beds and rooms:

(A16)	$\forall x\ \forall y\ (E(x,y) \rightarrow (B(x)\ \&\ R(y)))$	Relation "being installed" exists only between a bed and a room
(A17)	$\forall b\ \forall r\ (B(b)\ \&\ R(r)\ \&\ E(b,r) \rightarrow \forall s\ (E(b,s) \rightarrow s=r))$	A bed cannot be installed in more rooms

It is suitable to define relation N, representing the statement "patient p lies at time t in room r": $N(p,r,t) \equiv \exists b\ (L(p,b,t)\ \&\ E(b,r))$. Using it, we can simply formulate the next consequence of *(A14)* and *(A17)*:

(C10)	$\forall p\ \forall r\ \forall t\ (P(p)\ \&\ R(r)\ \&\ N(p,r,t) \rightarrow \forall s\ (N(p,s,t) \rightarrow s=r))$	A patient cannot lie in more rooms at the same time

The next group of axioms concerns relations among patients and rooms:

(A18)	$\forall r\ \forall t\ ((S(r)\ \&\ \exists p\ (N(p,r,t)) \rightarrow (\Phi(r,t) \vee M(r,t))))$	Every standard room which is not empty is either male room or female room
(A19)	$\forall r\ \forall t\ (R(r)\ \&\ \Phi(r,t) \rightarrow \forall p\ (P(p)\ \&\ N(p,r,t) \rightarrow \phi(p)))$	The patients placed in a female room are women
(A20)	$\forall r\ \forall t\ (R(r)\ \&\ M(r,t) \rightarrow \forall p\ (P(p)\ \&\ N(p,r,t) \rightarrow \mu(p)))$	The patients placed in a male room are men
(A21)	$\forall r\ \forall t\ (R(r) \rightarrow \sim(\Phi(r,t)\ \&\ M(r,t)))$	No room can be male and female at the same time
(A22)	$\forall r\ (S(r) \rightarrow R(r))$	Standard rooms are rooms

Under the term ***standard rooms*** the rooms are understood that belong to the standard section. Note that the demand of being either male room or female room holds only for those rooms. It does not hold for the other rooms, i.e. for the rooms in ICU, which are neither female rooms nor male ones. Naturally, the axiomatic system does not exclude the control systems that tend to organize male rooms and female rooms in ICU, too.

It is necessary to introduce axioms concerning time. The habitual PBS regime counts the events in days and therefore the system time of PBS could be discretized and possibly mapped as a sequence of integers. Nevertheless, we reflect the fact that all simulation programming tools of a good (and average) quality consider the system time as ordered interval of real numbers and we introduce it in such a way. It allows not only to reflect the fact that the patients can enter and leave the system in any time during a day, but also the fact that decisions can be demanded at any moment of the time interval during that the given PBS is considered as a system. A decision made in one day can essentially influence the constraints for another decision made in the same day later (see the last section of the present paper). Therefore, the axioms concerning time do not demand a patient who enters a bed to occupy it at least one day, but we formulate them in the following sense:

(A23)	$\forall p\ \forall b\ \forall t\ ((L(p,b,t)\ \&\ \exists u\ (u<t\ \&\ \forall w\ (u<w<t \rightarrow \sim L(p,b,w))) \rightarrow \exists v\ (v>t\ \&\ \forall w\ (t<w<v \rightarrow L(p,b,w))))$
(A24)	$\forall p\ \forall b\ \forall t\ ((\sim L(p,b,t)\ \&\ \exists u\ (u<t\ \&\ \forall w\ (u<w<t \rightarrow L(p,b,w))) \rightarrow \exists v\ (v>t\ \&\ \forall w\ (t<w<v \rightarrow \sim L(p,b,w))))$

The axiom *(A23)* tells that when a patient enters a bed it occupies it for a certain future non-degenerated time interval. This weak demand allows introducing any time scale into the system (in other words, the time unit does not need to be one day). The axiom *(A24)* is a certain counterpart to it, stating that when a patient leaves a bed it does not occupy it during a certain future non-degenerated time interval.

4 Basic Object-Oriented Representation of the Concepts

The contents of the concepts is exactly characterized by the axioms mentioned above and according to them the concepts were represented in the object-oriented mode offered by Simula (Dahl et al., 1968), (SIMULA Standard, 1989). According to the axioms *A11* and *A12*, there are three basic classes *patient*, *bed* and *room*, for which the only common set of properties is that their instances can enter lists. Entering lists is suitable for manipulation and evidence (namely in a form of queues). For the mentioned set of common properties, Simula offers a standard class called *link*. Therefore, the mentioned classes are subclasses of *link*.

The sex is not reflected by subclasses of class patient. It is sufficient to equip class *patient* by a Boolean attribute *woman*; it is a protected parameter and, therefore, every instance of *patient* gets a certain value of *woman* so that this value can be later read but not changed. The other attribute of class *patient* is *place*, pointing to the bed that the patient occupies, and *membership*, pointing to the room, to which the patient belongs. Oppositely, class *bed* has attribute called *content*, pointing to the patient who occupies it, and – similarly as *patient* – attribute *membership*, pointing to the room where the bed is installed. Similarly as attribute *women* of class patient, the attribute *membership* of class *bed* is protected but can be read.

The classes are equipped by a lot of methods that help to reflect the other axioms and to check whether they are not violated. For example, class *patient* has a pair of methods *take(b)* and *leave*, used for entering a bed *b* and for leaving the bed where the patient has placed. When a message of a form *take(b)* is sent to a patient *p*, it checks whether *p* is not placed at some bed, when bed *b* is free and – in case *b* is in a room that is a male room or a female room – *take(b)* checks the conformity of the *p*'s sex. When all is in order, *take(b)* assigns *b* to *p* for its attribute *place* and assign *p* to *b* for its attribute *content*, and then reads the value of *membership* of *b* and assign it to *p* for its attribute *membership*.

When a message *leave* is sent to a patient *p*, the checks are performed whether *place* of *p* points to a bed (when no bed is assigned to *p*, the value of place is equal to *none*, which is the identification serving for "nothing", offered by Simula). If it is so, i.e. if *place* of *p* points to a certain (existing) bed *b*, then *take* assigns *none* to *b* for its contents, puts *b* into the list of the free beds of the given room and assigns *none* to *p* for its attribute *place*. In case *place* was equal to *none*, then *leave* leads to a signal of error.

Both *take* and *leave* contain steps to keep evidence on the state of rooms. They are classified into three categories – *full*, *empty* and *half-empty* (rooms). In case a *half-empty* room is either male or female it can accept only a patient of a given sex, a *full* room cannot accept any patient and an *empty* room can accept any patient and according his sex it may turn to become *male* or *female*. All those manipulations are included inside *take* and *leave*, so that the rules for the ordinary dynamics of patients (called "life rules" in Simula) can be simply described by alternating messages *take(b)*, *hold(t)* and *leave*, where *hold(t)* is the standard method of Simula expressing the statement "wait until the system time increases by *t*". Therefore, the structures of the life rules of the classes of standard, serious and emergent patients (see the end of Section 1) can be described as follows (b_S is a bed in a standard section and b_I is a bed in ICU):

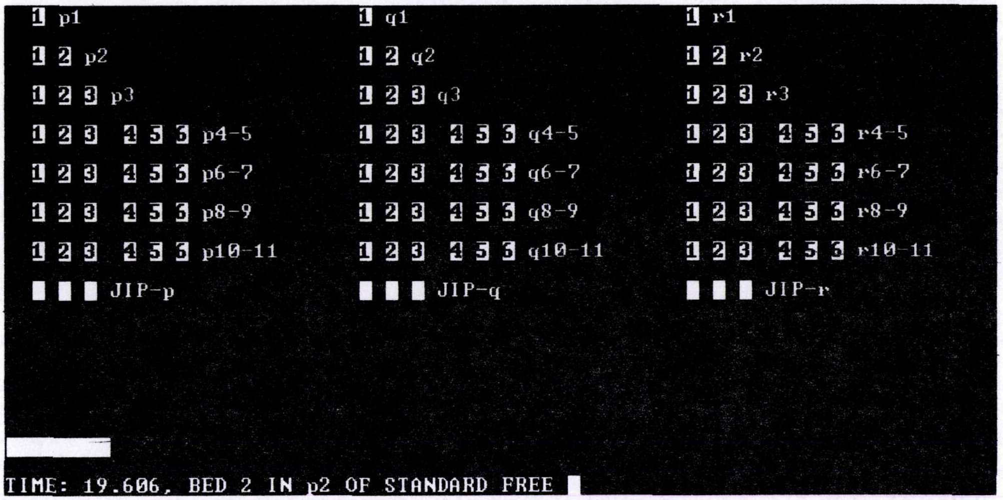
standard: *take(*b_S*); hold(...); leave;*

serious: *take(*b_S*); hold(...); leave; take(*b_I*); hold(...); leave; take(*b_S*); hold(...); leave;*

emergent: *take(*b_I*); hold(...); leave; take(*b_S*); hold(...); leave;*

5 Additions to the Object-Oriented Representation

Using Simula, the description of a PBS according to the way just outlined could be immediately translated into a simulation model. But the simulation model would run inside the electronics of the computer, while we would like to get some information about it. Therefore, the basic representation was completed, namely by internal collection of statistical data, by displaying them at the end of the simulation experiment, and by tools for continuous watching of the model run by the operator. Thus automatic accumulation of time, during which the concerned bed was occupied by a patient, was built inside procedures *take* and *leave* and at the end that value and relating data are not only displayed and recorded for the further manipulation but also processed to get information on the exploitation of the rooms.



```
1 p1                       1 q1                        1 r1
1 2 p2                     1 2 q2                      1 2 r2
1 2 3 p3                   1 2 3 q3                    1 2 3 r3
1 2 3  4 5 6 p4-5          1 2 3  4 5 6 q4-5           1 2 3  4 5 6 r4-5
1 2 3  4 5 6 p6-7          1 2 3  4 5 6 q6-7           1 2 3  4 5 6 r6-7
1 2 3  4 5 6 p8-9          1 2 3  4 5 6 q8-9           1 2 3  4 5 6 r8-9
1 2 3  4 5 6 p10-11        1 2 3  4 5 6 q10-11         1 2 3  4 5 6 r10-11
█ █ █ JIP-p                █ █ █ JIP-q                 █ █ █ JIP-r

TIME: 19.606, BED 2 IN p2 OF STANDARD FREE █
```

Figure 1: A monochrome image of a color snapshot from the visualization of a hospital model. The monochromatic projection does not allow showing the difference between the male and female rooms. The inscriptions like *p1*, *q4-5* or *JIP-r* are names of the rooms (JIP is a Czech equivalent for UIC).

In order to enable visual observing the model run, tools for emitting messages on any event were built into methods *take* and *leave* and tools for animation as well. The PBF is represented at the computer display as a map of the simulated PBS, namely as a structure composed of visual representations of beds and rooms. The dynamics of PBS is visualized by changing the colors of the beds; they are displayed as rectangles in blue, red and green color according to the *content* (man – blue, woman – red, empty – green). Inside the rectangles there is other (alphanumerical) information. By hitting keys, the modeller (operator) is able to change the speed of the computation, to interrupt the model run, to switch between the step-by-step regime and the continuous one and to reset the process of collecting statistical data.

In Figure 1, one can see an example of a snapshot taken at a model PBS composed of three parts of a similar structure.

6 Simulation Model M^a

As it was introduced in Section 2, the model M^a concerns a design, a (re)configuration or a (re)dimensioning of a PBF. The consequence is that in general no information for the model exists and it is necessary to count on the situation that all its parameters will be new. Therefore, the model M^a was constructed so that its user can construct PBF "from scratch" either by using keyboard and the display cursor, or using the computer mouse, and so he can "draw" the PBF structure. The image of the described PBF stepwise appears at the display in the same form as during the further animation (see e.g. the example in Figure 1). All the beds are considered empty and, therefore, their images at the screen shine in green. When the structure is complete one can store it in a file and be free of the lengthy drawing in the future experimenting (the third way to construct PBF is to do it according to the data recorded in such a file). Let that file be called *PBF-file*.

Then one describes the data that determine the duration of the phases of the patients' stay in the system, the rate of the patients entering, the ratios of the occurrences of the three types of the patients, the duration of the simulation experiments, etc. Pseudorandom values are admitted. The data can be also stored in a file and further simply read from it. In this case the dialog with the operator is similar as in case he puts new data – the only difference is that the data read from the file appear as offered values and the operator can either edit them or let them be accepted as they are offered (by simple hitting *enter* key).

Then the model starts and after the end of the simulation experiment statistical data are displayed and stored, informing on the exploitation of the components of PBF and of the queues of the patients who had to wait for places at beds.

7 Simulation Model M^b

It is necessary to be aware of the fact that the system S^b differs from S^a and that the manipulation with the simulation model is so different that also the model M^b has to differ from M^a. Let us illustrate it in a few details.

Model M^b is applied when the PBS exists and works and when the corresponding PBF contains patients. In such a situation, one cannot change PBF and one must even respect the states of the patients who are in it. The consequence is that it is meaningless for M^b to change its PBF but M^b cannot start with empty PBF but from the real situation of it.

Therefore, M^b does not need the initial procedures of a manual input of the data for drawing the structure of PBF (a possible editing the values coming from the PBF-file is very sufficient even for unexpected cases) but it needs a procedure for initiating the state of PBF. When the PBF appears at the computer screen, the operator is demanded to fill in PB: the first step of the dialog he should determine the value of the present time and then the beds are stepwise scanned; for each of them the operator has always to answer, whether it is empty or not, if not, what is the sex of the given patient, when the patient should leave the bed, of what sort he is and in case he is the serious one and the bed is SC the operator has to choose one of the variants – (α) the patient is being prepared before entering UIC and (β) the patient already returned from UIC.

8 Synthesis of the Models

Assume simulation model M^a is under preparation. It means that (a part) of the human society is an anticipatory system S^a that tries to know the future states of the designed PBD, i.e. that it tries to know some future states of itself. When this system is at a good professional level, it has to accept that the designed PBS will use something like model M^b. In other words, S^a must accept the fact that a part of it (may be a smaller part of the human society) will be an anticipatory system S^b, the behaviour of which is necessary to know during the design and during the application of M^a. A general proof of it was derived e.g. by Kindler (2000). It is based on the dilemma implied by the premise that M^b is neglected in the construction of M^a:

- either the application of model M^b by S^b would be unimportant and meaningless and its construction and implementation would be useless,
- or the model M^a would generate false information on the future states of S^a.

The consequence is that the model M^b has to be nested into the model M^a. Speaking in more details, the model M^a of the (part of the) human society S^a should reflect that S^a will contain not only a "conventional" PBS but together with it also a certain information processing element C (a computer) that will operate contemporaneously with the PBS and will interact with it – sometimes C will scan it to get values V_I for further computing and sometimes C will govern it with help of values V_O that it will produce. Let the algorithm for transformation from V_I to V_O be denoted A. Although a lot of the processes performed by C need not be reflected in M^a, the algorithm A has to be mapped in M^a in its detailed form (in other words – if A could be simplified or even neglected inside M^a it could be simplified or neglected in S^a, i.e. also in S^b and in the real PBS).

When A is based on simulation model M^b (i.e. when A functions so that it manipulates with M^b and performs experimenting with it), the manipulation with this model (and therefore this model, too) should be in details interpreted inside M^a, i.e. M^b must be nested inside M^a.

9 Ways of Synthesis of the Models

As it was mentioned e.g. by Kindler (2000) or more recently by Krivy and al. (2002), the implementation of nested simulation models is a difficult task. Simula preserved a good tool that was already offered by ALGOL 60 (Naur et al., 1963), namely nesting of blocks. It was later rejected by the world programming community (oriented to the so called modular programming, for which it seemed dangerous), but a long after it appeared as suitable and non-dangerous. In relation to the object-oriented programming, block nesting enables a very simple and convenient tool for nesting one (simulation) model into another, i.e. for nesting of PBSs, too.

Nevertheless, there are certain difficulties concerning the natural desire. If a model of a hospital (PBS, ...) is nested in another model of the same hospital (PBS, ...), we would like to describe both the models in similar languages, i.e. we would like to identify the components of the model (and the actions, properties, etc.) in the same manner, namely so as we identify their real components. For example, we would like use the words *bed2* (*room3*, classes *bed*, *room*, *patient*, etc.) in description of both the models, namely for the model components that reflect real beds, rooms, patients, etc. But when we accept it we are in a danger to mix the components of both the models – e.g. by a simple programming error (almost a "misprint") we could express a component existing in M^a to perform an action of using a bed existing in M^b.

Although the intention of the modeller was to express the event that a (real) patient is directed to a (real) bed, the error would be a true image of the event when a (real) patient is directed to lie down to a certain bed, but this bed is not a real bed of the same "world" where the patient exists, but a bed modelled at a computer; the computer exists in the same world as the patient but the bed exists only as some electronic phenomena inside the computer, to which we view as to an image of a bed. But this "imaginary" bed exists among other elements (beds, rooms, patients, etc.) of the imaginary world, these elements are in mutual relations and after several milliseconds of simulation, the mentioned small confusion can automatically cause an avalanche of other similar errors, a medley of the elements of both the models M^a and M^b and the model run often ends in a collapse. In such a situation one would need to start a reverse process to the avalanche, in order to reconstruct the progress that started from the first error to the final collapse, but the collapse does not allow doing it and, therefore, one cannot determine the first reason of it, i.e. the programmer's error.

Simula has tools for protecting its users against such errors, but those tools could seem to limit the model nesting, namely to use the same names for the elements, classes and activities of the models M^a and M^b and/or to copy the actual state of M^a into the initial state of M^b. One has to use sophisticated tricks to overcome those limitations. In case of the hospitals (PBS, ...), the following two ways appeared suitable; their difference can be characterized by means of the constructing M^b performed by a fictive modeler F who exists in S^a and operates the simulating computer:

(a) so called ***general style***; F does not know anything on his environment, he is modeled so that he has to inspect all elements of S^a as something that was hidden against him until the moment when M^b is to be constructed;

(b) so called ***special style***; F is modeled as knowing PBF, i.e. he always owns data on the structure of PBF (namely, he can use the data saved in PBF-file mentioned in Section 6), he constructs the basis of M^b according them and then he only scans all beds, all queues and states of all patients he met there and introduces copies of the patients into M^b.

The general style is really general and can be applied in constructing the nested models in any other cases, while the special style is oriented especially to the PBS. Nevertheless, it stimulates some ideas to be generalized outside of the boundaries of the hospitals (namely in domains of logistics, production and other services than the medical ones).

10 Conclusions and Further Development

Although both the techniques work at PCs, the authors orient their future investigation to "tailoring" the model for some special cases, namely:

(α) there are some PBSs with special scheduling of the patients' ingresses, like nuclear medicine sectors; in these sectors there is a less number of random events but the events in them are scheduled by complex scheduling rules;

(β) it would be suitable to construct some models of sort M^b so that reading databases of patients could replace the manual initial dialog.

Both the modifications can be implemented without programming obstacles, as Simula offers virtual methods, i.e. methods the content of which can be redefined without other interventions into the model. The only objective is to declare exact demands of the hospital sectors managers.

The models M^a and M^b were applied to anticipate the behaviour of some clinics in Prague (Czech Republic). The scheme presented in Figure 1 represents one of them, namely a clinic structured into three almost equivalent sections, each of which having its UIC. The horizons of the tests performed by application those models are very large, including relations between the sections and therapeutic streams and placing the patients in given rooms. The nesting of M^b inside M^b is being prepared for dimensioning the sections after reorganizing the clinic, when it will be controlled by the help of a computer that will be able to simulate. The nesting is a so hard task that nowadays it is tested whether the model behaves duly.

Acknowledgement
This work was supported by the Ostrava university research scheme MSM6198898701.

References

Dahl Ole-Johan, Myhrhaug Bjørn, Nygaard Kristen (1968) Common Base Language. Norsk Regnesentralen, Oslo (1st ed); 2nd ed. 1972, 3rd ed. 1982, 4th ed. 1984.

Kindler Eugene (2000) Chance for SIMULA. ASU Newsletter **26**, 2-26.

Krivy Ivan, Kindler Eugene, Tanguy Alain (2002) Software for Simulation of Anticipatory Production Systems. International Journal of Computing Anticipatory Systems **11**, 320-335.

Naur Peter et al. (1963) Revised Report on the Algorithmic Language ALGOL 60. Communications of the Association of Computing Machinery **6**, 1-17.

SIMULA Standard (1989). SIMULA a.s., Oslo.