Virtual Stochastic Sensors for Hybrid Systems: Mutual Influence between Continuous and Discrete System Parts

Claudia Krull[1], Graham Horton[1]
[1]Otto-von-Guericke-Universität Magdeburg
claudia.krull@ovgu.de

Virtual Stochastic Sensors can reconstruct the unobservable behavior of discrete stochastic systems based on time-stamped output protocols. Previously we extended the underlying modeling paradigm by continuous reward values. Measurement protocols of these values were then used for the reconstruction of the discrete model behavior. The current paper goes one step further by enabling the continuous model part to influence the discrete behavior through arbitrary guard functions, which can enable or disable activities in the discrete model part. Thereby making the model truly hybrid. Two examples from medicine and the renewable energy sector are taken to exemplify the potential and current limitations of the paradigm. Solely based on infrequent measurements of one continuous value, are we able to reconstruct likely behavior of a complex hybrid model. This enlarges the application domain of Virtual Stochastic Sensors to many real life problems with hybrid properties.

1 Introduction

Most real world systems are of hybrid nature, exhibiting discrete as well as continuous behavior. Most often these two influence each other, requiring a powerful modeling paradigm to capture these dynamics accurately. To complicate matters further, the systems are often only partially observable, due to infrequent measurements, sparse sensor coverage or actually hidden parts.

Fluid stochastic Petri nets (FSPNs) [1] and stochastic reward nets (SRNs) [2] are two complex hybrid modeling paradigms. However, one has to be able to observe and analyze all processes involved in the real system, in order to accurately represent the system in one of these paradigms. Simulation and behavior prediction of partially observable systems becomes more and more vague, as the hidden fraction increases. We therefore propose using Virtual Stochastic Sensors to perform actual reconstruction of past behavior, based on measurements obtained from the real system.

Virtual Stochastic Sensors have been successfully applied to behavior reconstruction of discrete stochastic systems ([3]). The underlying modeling paradigm Hidden non-Markovian Models has been extended to incorporate continuous reward values ([4]). As of now, we were not able to model influence of the continuous on the discrete model part.

In this paper we extend the paradigm to truly hybrid systems, by enabling the values of the continuous quantities to influence the enabling of discrete model transitions. This is done by incorporating the well-known concept of guard functions, which can be arbitrary logical expressions based on the current model state.

Two examples are taken to illustrate the modeling power and application potential of the extended modeling paradigm. One is taken from the renewable energy sector, the other from the medical domain. Using VSS we are able to reconstruct the dynamics of a small solar thermal energy system and its dependence on optional external heating, as well as the behavior of a type-one diabetic based on his blood glucose protocol and thus his adherence to his diet and injection schedule.

The proposed extension opens up a new class of systems for behavior reconstruction via VSS, making the technology applicable to real world problems of various types.

2 Related and Previous Work

2.1 Hybrid Modeling Paradigms

There exist a number of hybrid modeling paradigms. The two presented here are both based on stochastic Petri nets (SPNs). They are chosen as representatives, because the HnMM paradigm borrows heavily from SPNs. A comprehensive introduction to stochastic Petri nets can be found in [5].

2.1.1 Fluid Stochastic Petri Nets

Fluid stochastic Petri nets were first introduced in [1] and have been considerably extended since then [6]. They extend SPNs by adding fluid places and fluid arcs. Fluid places can contain a positive real-valued amount of fluid. Fluid arcs connect transitions and fluid places. When one such transition is enabled, the fluid arc can fill or drain the fluid place with a constant flow rate. Later on, flush-out arcs were introduced that can change a fluid level instantaneously. Besides representing actual fluids, FSPNs are widely used in performance and reliability modeling, where the measures of interest are depicted as a fluid level. They are also used to abstract a potentially large number of discrete tokens via a continuous value. FSPNs enable the discrete and continuous model part to influence each other.

2.1.2 Stochastic Reward Nets

Stochastic reward nets (SRN) extend SPNs by adding external continuous measures [2]. Rate rewards may change these values continuously, based on the current system state. Impulse rewards can change the values instantaneously upon the completion of a system activity (firing of a transition). However, the value of the reward measures do not influence the behavior of the discrete system. Like FSPNs, SRNs are often used in reliability and performance analysis.

2.2 Virtual Stochastic Sensors

Virtual stochastic sensors have been introduced in [7] and can reconstruct unobserved behavior of discrete stochastic systems, based on observable system output. VSS are related to virtual sensors, which are used to determine a quantity of interest that is not or not easily measurable from values which are easier to obtain [8]. The quantity of interest can often be computed directly from the sensor readings, due to a well-formed relationship. In VSS, this relationship is stochastic, and can be expressed using so-called Hidden non-Markovian Models, which were first introduced in [9]. In HnMM the unobserved system is described by a discrete stochastic model, and the measurable quantities are represented by output symbols, which are emitted via a second stochastic process based on the system states or transitions.

HnMM in turn are based on the well known Hidden Markov Models [10]. These doubly stochastic processes use a discrete-time Markov chain (DTMC) [11] as hidden model, and generate an output symbol in every step depending on the current state of the DTMC. HnMM enable the state transitions to be described by arbitrary continuous distribution functions, and allow the concurrent enabling of multiple non-Markovian activities. This enables behavior reconstruction for a much larger class of systems compared to HMM and extensions [10].

In [4] hidden non-Markovian reward models (HnMRM) were introduced, which extend the discrete HnMM paradigm by continuous reward measures, as they are used in SRNs. This broadens the application range to include some hybrid systems. However, the discrete model part influences the continuous part, but not vice versa. The current paper closes this gap, and extends the paradigm to truly hybrid systems.

2.2.1 Proxel-based Behavior Reconstruction

There are three main behavior reconstruction tasks defined for HMM and other related paradigms. *Evaluation* determines the probability of a given model to have produced a given output sequence. It is most often used to determine which of several models most likely produced a given sequence, and therefor closely resembles the system having produced the observations. *Decoding* determines the most likely sequence of system states to have produced a given output sequence, thereby helping to understand the unobserved dynamics of the system. *Training* tries to determine a likely model given an observed output sequence, but as opposed to evaluation by manipulating the model parameters, e.g. through iteratively refining an initial model.

The two behavior reconstruction tasks *Evaluation* and *Decoding* for HnMRM can be performed using the so-called Proxel method. For a detailed introduction to Proxels see [12, 13]. There exists currently no general solution to the third task of Training HnMM. However, it can be solved for subclasses and some special cases [14]. Proxels are a state space-based analysis method to iteratively discover all possible system developments in discrete time steps and quantify them with their respective probabilities. Proxels can be used to mimic the HMM reconstruction algorithms by iteratively generating likely paths that could have produced the given output sequence. Proxels are ideal for path reconstruction and can be applied to various types of HnMM, including HnMRM [4, 7, 14]. In this paper we will adapt the HnMRM behavior reconstruction algorithm to hybrid models.

3 Hybrid Hidden non-Markovian Models

In this section we will formally define hybrid hidden non-Markovian models (HHnMM) and briefly describe the corresponding behavior reconstruction algorithm. As basis we will take the Hidden non-Markovian reward models (HnMRM) as described in [4]. These use rate rewards associated to the discrete states and impulse rewards associated to the transitions to model the influence of the discrete on the continuous model part. The influence of the continuous on the discrete model part will be modeled through guard functions. Guard functions (GF) are used in some SPN paradigms to express complex marking-dependent enabling rules for transitions. A transition with a guard function is disabled, if the GF evaluates to false. Thus by including terms depending on the continuous values in the GF, we can make the discrete part of the model depend on the continuous quantities. One can for example disable a transition if a particular continuous quantity leaves a given range, or enable an immediate transition when a threshold value is crossed, and thus invoke an instantaneous state change.

3.1 Formal Definition

The formal definition of HHnMM uses the state space of a discrete stochastic model and the values of the reward measures as hidden model part. An HHnMM is formally defined as a 6-tuple $HHnMM = (S, TR, A, \Pi, \vec{\rho}, rr)$ with trace O and path Q.

- S is the set of N discrete states of the system.
- TR is the set of state transitions, where each $TR_i = (dist, gf, ir, aging)$ can represent multiple state changes; *dist* describes the continuous probability distribution function that governs a state transition; guard function gf is a boolean expression defining a complex enabling rule based on the current system state, default is $true$ if not specified; $ir(\vec{\rho}, s, s')$ describes the changes to the vector $\vec{\rho}$ due to impulse rewards when the transition fires, changing the system state from s to s'; *aging* is set to true, if the transition has a race age memory policy (i.e. it retains the time elapsed since activation, when it is disabled before firing, and resumes the countdown when re-enabled, without re-sampling the distribution).
- $A = \{a_{ij}\}_{NxN}$ describes the complete state transition behavior and a_{ij} is the set of transitions that can cause the state transition from S_i to S_j .
- $\Pi = (\pi_1, ...\pi_N)$ is the vector containing the initial probabilities of all discrete states and π_i is the probability of the system to be in state S_i at time 0.
- $\vec{\rho} = (\rho_1, ...\rho_M)$ is the vector containing the initial values of all continuous quantities.
- The rate rewards are defined by the function $rr(\vec{\rho}, s)$. It describes the changes to the vector $\vec{\rho}$ due to rate rewards in the form of a differential equation $\frac{d\vec{\rho}}{dt}$ depending on the state s.
- O is a sequence of measurements o_t of one (or more) of the continuous quantities in $\vec{\rho}$ with time

stamps e_t and often referred to as a trace. withoput loss of generality we assume that ρ_1 is the sampled value.

- Q is a sequence of system states s_t with time stamps e_t and often referred to as a path.

The model $\lambda = \{A, \Pi, \vec{\rho}, rr\}$ completely determines the dynamic behavior of the system and is therefore often used as its representative.

We now have to adapt the behavior reconstruction algorithm introduced in [4] to include complex enabling rules in the form of guard functions.

3.2 Behavior Reconstruction of HHnMM

In this section we will describe the HHnMM behavior reconstruction algorithm adapted from ([4]). The general idea of the Proxel-based behavior reconstruction algorithm is to follow all possible system development paths in discrete time steps; on-the-fly determining possible discrete system states, resulting continuous quantities and the probability of each path. When a protocol entry o_t occurs within a time step, the respective continuous value ρ_1 of each path is compared to o_t. The path is kept, if the difference is smaller than a given threshold ε. Otherwise, the path will be deleted, and not considered further.

The newly included guard functions influence the enabling of transitions and therefore need to be checked, when determining whether a given transition is currently enabled. The following verbal description of the HHnMM behavior reconstruction algorithm highlights where that is necessary.

1. The initial Proxel representing the initial state of the model is created in t_0.
2. For each time step t_k ($k = 1..k_{max}$)- Loop until there are no more Proxels in time step t_{k-1}.
3. Get the next Proxel P from time step t_{k-1}.
4. If there is a protocol entry with a time stamp between t_{k-1} and t_k, check whether the path is valid (Proxel reward value ρ_1 is close to protocol reward value o_t)
5. If there are immediate transitions enabled **(their guard function evaluate to** *true*), the Proxel represents a vanishing state,

   add Proxels representing these immediate state changes to t_{k-1}, modifying the reward values according to the transitions impulse rewards.

   If the path is still invalid, even after the impulse rewards, delete the Proxel.
6. If the state is not vanishing, modify the Proxels reward values according to the current rate reward, check path validity again
7. For all timed transitions that are enabled in the Proxels current state **(whose guard functions evaluate to** *true***)**

   add Proxels representing these state changes to t_k, modifying the reward values according to the transitions impulse rewards.

   If the path is still invalid, even after the impulse rewards, delete the Proxel.
8. If there is a non-zero probability of no state change happening, and the current path is valid, add a Proxel, representing the case, that the system remains in the discrete state to t_k.

Determining the validity of a path needs to be done, when the time stamp of a protocol entry lies within the currently considered time step. The distance of the protocol value and the path reward value is tested at the beginning and at the end of the time step, i.e. before and after the rate rewards have been added; before and after a state change happens, i.e. before and after considering the impulse rewards. If the Proxels reward value is within ε of the protocol reward value at any of these instances, the path is deemed valid.

At the end of the reconstruction (after processing the complete protocol) The result of the algorithm is a list of system development paths, that could have produced the protocol with non-zero probability. These paths then need to be analyzed further to answer the questions posed for the given system. For the solar thermal system, the amount of external heating energy needed could be determined, giving an indication on whether the equipment performs to the given standards. One could also search for the most likely cause of an unusually high need for external energy, be it an unusually large or an untimely usage profile. For the diabetic, one could determine how well the patient adheres to his dietary and medication plan, answering the following questions: Did he forget an insulin injection? How often did he eat a snack? The post-processing analysis can be done on the most likely path(s) or on the path(s) with the smallest divergence from the protocol. The following section shows some results for the two example systems.

4 Experiments

In this section we will demonstrate that Virtual Stochastic Sensors can reconstruct unobserved behavior from given protocols of hybrid systems. In order to test the validity of the reconstructed system behavior, we used simulation models to generate output protocols, also recording the actual model behavior. This will enable a direct comparison between the actual system behavior and the reconstructed path.

4.1 Demonstrating Path Reconstruction - Diabetes Example

The first experiment is conducted with the same application example chosen in [4]. The model describes the dynamics of the glucose insulin metabolism of a type-one diabetic, who can not produce his own insulin and has to inject insulin regularly. The interaction dynamics of the insulin level (*IL*) and glucose level (*GL*) have been taken from [15]. A graphical representation of the HHnMM can be seen in Figure 1 in the form of an SPN with impulse rewards. The rate rewards are depicted in the form of differential equations, since these are the same for all discrete states. We assume the patient to have two distinct states *hungry* and *full*, where after eating, he remains in state *full* for a time period that is determined by a normal distribution with mean $\mu = 7.5h$ and a standard deviation of $\sigma = 1h$. In state hungry he will again remain for a normally distributed time period with parameters $\mu = 0.5h$ and $\sigma = 0.1h$. This combination leads to eating intervals with an average length of eight hours. When the patient eats, his blood glucose level is increased by a fixed amount (1000*units*), which is an idealization. While eating, he should inject an also fixed amount of insulin (3000*units*).

The patient sometimes forgets to inject his insulin, thus not increasing his blood insulin level. Furthermore, when the blood glucose level drops below a certain threshold (5000*units*), the patient eats a snack, increasing his glucose level by a smaller amount (200*units*). This dependence of the discrete model on the continuous measures is expressed by guard functions, making the model truly hybrid.

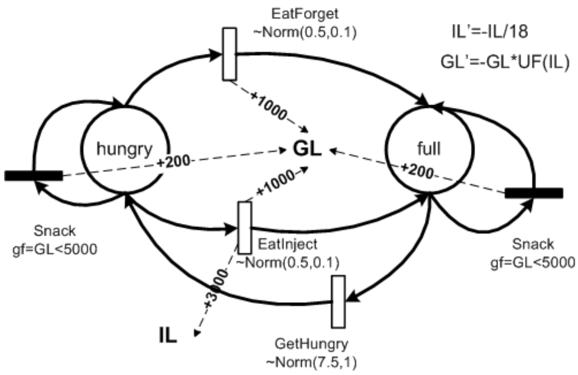

Figure 1: A graphical representation of the insulin glucose metabolism model of a diabetic in the form of a state space with associated rewards

We now assume, that the patient measures his blood glucose level at regular intervals and records the results in a protocol. From such a protocol we want to deduce, how often he ate snacks, and how often he forgot to inject his insulin dose. This could help a doctor to determine how well the patient adheres to his dietary and medication plan. Figure 2 shows a measurement protocol (trace), which was produced by a simulation of the patient dynamics.

The reconstruction algorithm was applied to the trace with 32 entries, which represented a time span of 100*h*. Figure 3 (left) shows the simulated system path that produced the trace in Figure 2, which was recorded during the simulation.

The reconstruction algorithm resulted in a list of 45050 possible paths that could have produced the given trace ($\varepsilon = 60$). The computation time of the reconstruction was 11 seconds. Figure 3 (right) shows the most likely of these reconstructed paths. A comparison with the original system path shows that the HHnMM reconstruction algorithm was able to almost exactly reproduce the simulated behavior that led to

time stamp	GL
2.36	5581
5.82	5109
9.27	6131
12.11	5827
14.82	5577
18.00	6191
20.83	5858
23.86	5559
26.30	6340
29.41	5923
32.43	5599
35.44	6271
38.30	5879
41.60	5518

Figure 2: Protocol of regular glucose level measurements with time stamps

time stamp	transition
6.79	Snack
8.69	Snack
8.73	EatForget
15.94	EatInject
26.14	EatInject
34.70	EatInject
42.65	EatInject
48.58	EatInject
58.82	EatInject
65.29	EatInject
72.95	EatInject
80.00	EatInject
86.42	EatInject
92.56	EatInject

time stamp	transition
7	Snack
9	Snack
10	EatForget
16	EatInject
25	EatInject
34	EatInject
42	EatInject
50	EatInject
58	EatInject
66	EatInject
73	EatInject
79	EatInject
87	EatInject
94	EatInject

Figure 3: Eating and injection times in original system path (left) and reconstructed system path (right)

the observed trace. In particular, it correctly reproduced the two snacks and the one forgotten injection. The differences in the time stamps are mostly within the discretization accuracy of $1.0h$.

We can conclude, that the HHnMM behavior reconstruction algorithm was successfully applied to a modified version of the glucose insulin metabolism model, correctly reproducing the patients deviations from his dietary and medication plan. On a more abstract level, we were able to reconstruct behavior of a hybrid system with complex dynamics within the continuous part, described by differential equations. The state space of the model is small, and the trace analyzed short, thus the runtime is not an issue here. However, memory and computation time are strongly influenced by model parameters, as shown in [4].

4.2 Demonstrating Path Reconstruction - Solar Thermal Heating

The second experiment was conducted with an application taken from the field of renewable energy. We have built a simple idealized model of a small home solar thermal system, using simple dynamics and constant rewards, to demonstrate the algorithms ability to handle a larger state space and a differentiated rate reward structure. A graphical representation of the model is shown in Figure 4, again using SPN notation for the discrete model part. Here we can easily distinguish three independent discrete system parts, that only interact through the continuous measure.

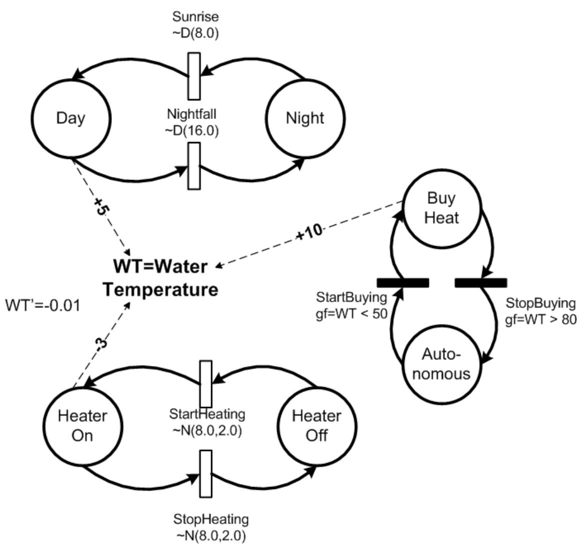

Figure 4: A graphical representation of the solar heater dynamics

The central element of the small solar heating system is the water tank with its water temperature WT as central attribute. The water temperature can be increased by the solar thermal collectors on the roof, or by an external heat source that has to be paid for. The temperature decreases, when the home heating system is turned on. The water temperature is bounded between $20°C$ and $90°C$. The first discrete system part is the rising and setting of the sun, which is abstracted to a fixed $16h$ period of daytime and an fixed $8h$ period of night. When the sun is up, the water temperature increases linearly by $5°C/h$. The second discrete part is the random usage of the home heating system. The heater switches between *On* and *Off* in normally distributed intervals with parameters $\mu = 8h$ and $\sigma = 2h$. When the heater is turned on, the water temperature decreases linearly by $3°C/h$. The third element is the addition of external heating, which is switched on, when the water temperature falls below $50°C$ and switched off, when it reaches $80°C$. When the external heating is turned on, the water temperature increases linearly by $10°C/h$.

We assume that the temperature of the water tank is recorded in a protocol of regular measurements. An example of such a trace is shown in Figure 5.

time stamp	WT
0	50
5.03	60.02
10.01	62.76
15.05	59.94
19.84	60.4
24.86	84.26
29.95	90
35.04	83.96
39.96	69.15
44.77	57.47
49.68	65.8

Figure 5: Protocol of regular water temperature measurements with time stamps

The reconstruction algorithm was applied to a trace with 132 entries, which represented a time span of $1000h$. In Figure 6 (left) shows the simulated system path that produced the trace in Figure 5, reduced to the periods of external heating needed.

time stamp	transition
45.27	StartBuying
48.69	StopBuying
358.91	StartBuying
361.62	StopBuying

time stamp	transition
46	StartBuying
50	StopBuying
359	StartBuying
362	StopBuying

Figure 6: External heating periods in original system path (left) and reconstructed system path (right)

The reconstruction algorithm resulted in a list of 115 possible paths that could have produced the given trace ($\varepsilon = 2.0$). The computation time needed for the reconstruction was 3 minutes. Figure 6 (right) shows the most likely of these reconstructed paths. The HHnMM reconstruction algorithm correctly reproduced the two periods of external heating needed. The differences in the time stamps are mostly within the discretization accuracy of $1.0h$. All but one of the random heating periods were also reconstructed correctly, but are not depicted here.

In conclusion, we were again successful in reconstructing the system behavior that lead to a given protocol. The solar heater system differs from the diabetes example in that the rate rewards are all linear, but the discrete model state space is larger, containing not two but eight discrete states, with two concurrent non-Markovian transitions being enabled simultaneously.

5 Conclusion and Outlook

The paper introduced a method to reconstruct unobserved behavior of hybrid systems from measurement protocols of the continuous variables. These virtual stochastic sensors (VSS) use hybrid hidden non-Markovian models (HHnMM) to represent the hybrid systems. HHnMM contain a complex reward structure to model influences of the discrete on the continuous part and guard functions that can disable transitions based on reward values to model the influence of the continuous on the discrete part. A medical example illustrates the potential complexity of the reward measures, and an example from the renewable energy sector shows a collection of discrete systems that interact through a continuous system. This extension of the previous hidden non-Markovian reward models (HnMRM) to truly hybrid systems extends the possible application fields of VSS for behavior reconstruction.

The current paper only demonstrates the ability of VSS to reconstruct a systems behavior solely based on infrequent measurements of continuous measures. The runtime and memory complexity of the analysis are not yet satisfactory, since they are highly dependent not only on the model but also on method parameters. Future research will focus on guidelines and automatic processes to help optimize the algorithm parameters for a specific application. Starting from the academic example used here we want to research actual applications in the renewable energy field, since this is a highly dynamic area and most of the systems encountered there are inherently hybrid.

References

[1] K. S. Trivedi and V. G. Kulkarni. *FSPNs: Fluid stochastic Petri nets.* In: Application and Theory of Petri Nets 1993, Publisher: M. Ajmone Marsan, Lecture Notes in Computer Science, Springer Berlin Heidelberg, p. 24-31, 1993.

[2] J. Muppala, G. Ciardo and K. S. Trivedi. Stochastic Reward Nets for Reliability Prediction. *Communications in Reliability, Maintainability and Serviceability*, vol. 1, no. 2, p. 9-20, 1994.

[3] C. Krull, G. Horton, B. Denkena and B. Dengler. *Virtual Stochastic Sensors for Reconstructing Job Shop Production Workflows.* Proceedings of the 8th EUROSIM Congress on Modelling and Simulation, in Cardiff, UK, September 2013.

[4] C. Krull and G. Horton. *Hidden non-Markovian Reward Models: Virtual Stochastic Sensors for Hybrid Systems.* Proceedings of the Winter Simulation Conference, in Berlin, Germany, p. 224:1–224:12, 2012.

[5] F. Bause and P. S. Kritzinger *Stochastic Petri Nets.* Vieweg, Germany, 2002.

[6] B. Tuffin, D. S. Chen and K. S. Trivedi. Comparison of Hybrid Systems and Fluid Stochastic Petri Nets. *Discrete Event Dynamic Systems*, vol. 11, no. 1-2, p. 77-95, 2001.

[7] C. Krull, R. Buchholz and G. Horton. *Virtual Stochastic Sensors: How to gain Insight into Partially Observable Discrete Stochastic Systems.* Proceedings of the 30th IASTED International Conference on Modelling, Identification and Control, Innsbruck. Austria, February 2011.

[8] P. H. Ibargüengoytia and A. Reyes *Constructing Virtual Sensors Using Probabilistic Reasoning.* In: MICAI 2006: Advances in Artificial Intelligence, Publisher: A. Gelbukh and C. A. Reyes-Garcia, Lecture Notes in Computer Science, Springer Berlin Heidelberg, p. 218-226, 2006.

[9] C. Krull and G. Horton. *HIDDEN NON-MARKOVIAN MODELS: FORMALIZATION AND SOLUTION APPROACHES.* Proceedings of 6th Vienna Conference on Mathematical Modelling, Vienna, Austria, p. 682-693, February 2009.

[10] G. A. Fink *Markov Models for Pattern Recognition.* Springer, Berlin, Heidelberg, 2008.

[11] G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi *Queuing Networks and Markov Chains.* John Wiley & Sons, New York, 1998.

[12] S. Lazarova-Molnar. *The Proxel-Based Method: Formalisation, Analysis and Applications.* Dissertation, Otto-von-Guericke-University Magdeburg, Germany, November 2005.

[13] G. Horton. *A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times.* Proceedings of the European Simulation Symposium 2002, in Dresden, Germany, p. 129-136, October 2002.

[14] R. Buchholz. *Conversive Hidden Non-Markovian Models.* Dissertation, Otto-von-Guericke-University Magdeburg, Germany, Mai 2012.

[15] J. Hartlove, D. Shaffer and S. Ragan. *Glucose-Insulin Model. A core learning goals activity for science and mathematics*, The Maryland Virtual High School. 2001.