

Specification and Verification of Radiation Therapy System with Respiratory Compensation using Uppaal

Tomas Krilavičius, Kaiyu Wan, Kevin Lee, and Ka Lok Man

Abstract—The goal of radiation therapy is to give as much dose as possible to the target volume of tissue and avoid giving any dose to a healthy tissue. Advances of the digital control allow performing accurate plans and treatments. Unfortunately, motion compensation during the treatment remains a considerable problem. Currently, a combination of the different techniques, such as gating (restricting movement of patient) and periodic emission are used to avoid damaging healthy tissue. This paper focuses on systems that completely compensate respiratory movement (up to certain limit) and start by investigating adequacy of the existing hardware and software platform.

In this paper a radiation therapy system consisting of a HexaPOD couch with 6-degrees movement, a tracking camera, a marker (markers) and a controller is modeled. A formal un-timed model was evaluated and found to be insufficient to completely determine adequacy of the system to compensate respiratory motion. Therefore, un-timed model was extended to include time and investigated. It provides more information than un-timed model, but does not answer all interesting question. Therefore, based on the results further research directions are sketched.

Index Terms—simulation, verification, formal methods, radiation treatment, quality assurance

I. INTRODUCTION

THE goals of the radiation therapy is to give as much dose as possible to the target volume of tissue and avoid giving any dose to a normal tissue. Advances of the computer-based control allow planning and performing accurate plans and treatments, however motion compensation during treatment remains a considerable problem. Different techniques to cope with such problem are analyzed in [1]. Usage of gating combined with external surrogates is overviewed in [2]. However, most of the research models and try to predict movement of the tumor, e.g. [3]–[5]. This paper, on the other hand, is interested in modeling hardware and software, which is supposed to conform to the requirements, i.e. process images and move precisely and fast. Formal methods are used for such analysis, because they provide means for rigorous modeling

and analysis of diverse systems. The main reasons of the formal methods' popularity are the following.

- **Unambiguous models.** Formal modeling languages allow defining systems unambiguously, because syntax and semantics are defined formally, and those languages include means to define non deterministic and stochastic behavior precisely, too. Moreover, for the same reasons, unambiguous refinement and code generation techniques can be applied.
- **Strict analysis techniques.** Because models are defined using languages with strict semantics, rigorous reasoning about models is possible. E.g., model checking, theorem proving and specifically designed algorithms can be used.

Quite a few techniques and tools were defined over the years, e.g. process algebras [6]–[11], timed automaton [12], hybrid automaton [13], SPIN [14] and Uppaal [15] tools and a lot more, see [16], [17] for a wider overview. Successful application of formal techniques is reported in different areas, e.g. automotive industry [18], electronics [19], industrial devices control [20] and other.

This paper investigates applicability of timed automaton [12] and Uppaal tool [15] for the design and functional analysis of a radiation therapy system consisting of a HexaPOD couch with 6-degrees movement, a tracking camera, a marker (markers) and a controller. Uppaal is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types and other convenient constructions [15]. In [21] an un-timed version of the model was presented. However, the model is too abstract to determine adequacy of the system for a respiratory motion compensation task. Therefore, it was extended to include some timing properties and analyze some functional properties in [21]. In this paper timed model and timing aspects are presented in detail. Moreover, functional properties, i.e. absence of deadlocks, liveness and safety, are analyzed.

In Section II a detailed description of the radiation treatment system is provided. Then Uppaal and timed automaton in Section III are concisely introduced. In Section IV a Uppaal model of the radiation treatment system is presented, some of its properties are checked, and its applicability to further analysis is discussed. Future plans and conclusions are discussed in Section V.

T. Krilavičius is with Faculty of Informatics, Vytautas Magnus University, Informatics fac., Kaunas, Lithuania, and Baltic Institute of Advanced Technology. E-mail: (see <http://www.surface.lt/krilaviciust>).

Kaiyu Wan is with Xi'an Jiaotong-Liverpool University, Dept. of Computer Science and Software Engineering, 111 Ren'ai Road, Suzhou, Jiangsu 215123. E-mail:kaiyu.wan@xjtlu.edu.cn

K. Lee is with School of I.T, Murdoch University, Australia. E-mail:kevin.lee@murdoch.edu.au

K.L. Man is with Xi'an Jiaotong-Liverpool University, Dept. of Computer Science and Software Engineering, 111 Ren'ai Road, Suzhou, Jiangsu 215123, and Myongji University, South Korea. E-mail:ka.man@xjtlu.edu.cn

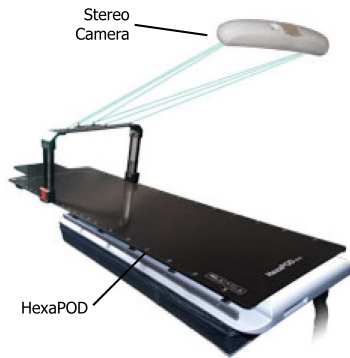


Fig. 1. Radiation Treatment System.

II. RADIATION TREATMENT SYSTEM

Radiation treatment system under analysis¹, depicted in Fig. 1, consists of the following components:

Patient Setup Couch is used to position the patient for the treatment, in our case the HexaPOD couch [22], [23].

External Radiation Beam Source, usually produced by a *medical linear accelerator*, in short, *linac*. In the current stage of our study it is not important, because behavior of the couch, the tracking device and the controller are analyzed.

Tracking Device provides information about the position of the patient. Different means and techniques can be used to perform it, see [1] for the details. A system with a stereo camera is modeled. In this paper hardcoded trajectories are used instead of dynamic input, and therefore, it is omitted.

Controller is a system, that controls the treatment process, in our case the controller uses information provided by the treatment plan and the HexaPOD response to control it.

A. Experiments with HexaPOD Couch

Technical documentation of the HexaPOD device does not provide detailed documentation of its behavior when it is used continuously, not just to move a patient into a specified position. Usually, when a new position is provided, it starts to move towards it accelerating with 5.5m/s^2 acceleration until it reaches 7.6m/s (instead of the stated 8mm/s velocity. Then, when 5mm are left to the target, it starts decelerating with 5.5m/s^2 . In case, when distance to the target is less than 5mm , HexaPOD accelerates until the middle of the interval and then decelerates until it reaches target and stops.

Based on these experiments and expected breath movement timing properties of the model and testing trajectories can be defined.

¹There is a diversity of radiation treatment systems, see [1] for overview of the systems relevant to this study. However, here we define just a selected setup.

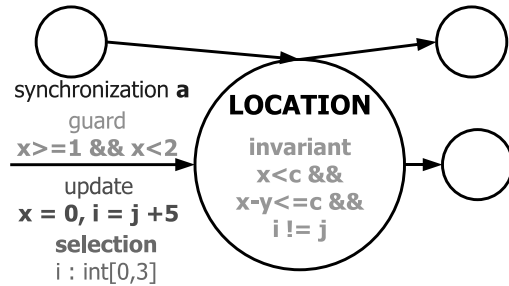


Fig. 2. Timed Automaton.

III. TIMED AUTOMATON AND UPPAAL

Timed automaton [12] is one of the most popular techniques for modeling and analysis of the real-time systems. A version of automata used in Uppaal [15] is presented. Uppaal is an integrated tool environment for the modeling, simulation and verification of (complex) real-time systems. It is well-suited for systems that can be modeled as a collection of non-deterministic processes with finite control structure and real-valued clocks, communicating through channels or shared variables.

Definition 1. Let $\mathcal{C} = \{x, y, z, \dots\}$ be a set of clocks and $\mathcal{B}(\mathcal{C})$ is the set of clock restrictions of the form $g, g_1, g_2 ::= x \bowtie c | x - y \bowtie c | g_1 \wedge g_2$ with $x, y \in \mathcal{C}, c \in \mathbb{N}$ and $\bowtie \in \{\leq, <, =, >, \geq\}$.

Definition 2. A timed automaton is called as a finite directed graph $A = (L, l_0, \mathcal{A}, E, I)$ over \mathcal{C} and $\mathcal{B}(\mathcal{C})$, where

- L is a finite set of locations;
- $l_0 \in L$ is the initial location;
- \mathcal{A} is a finite set of action names;
- $E \subseteq L \times \mathcal{B}(\mathcal{C}) \times \mathcal{A} \times 2^{\mathcal{C}} \times L$ is a finite set of edges, and $I : L \rightarrow \mathcal{B}(\mathcal{C})$ assigns invariants to locations.

$l \xrightarrow{g, a, r} l'$ is written instead of $(l, g, a, r, l) \in E$. l is called the source location of the state, g is the guard, a is the action, r is the set of clocks to be reset and l' is the target location. Timed automata can be represented as in Fig. 2. Locations are depicted as nodes of the graph, and the initial location is usually marked with a double circle. Transitions are depicted by arrows.

Definition 3. Let $A = (L, l_0, \mathcal{A}, E, I)$ be a timed automaton over a set of clocks \mathcal{C} . The timed transition system $T(A)$ generated by A is defined as $T(A) = (S, Act, \xrightarrow{tr})$, where:

- $S = L \times (\mathcal{C} \rightarrow \mathbb{R}_{\geq 0})$ is a set of states (l, v) , where l is a location of the timed automaton and v is a clock valuation that satisfies the invariant of l ;
- $Act = \mathcal{A} \cup \mathbb{R}_{\geq 0}$ is the set of labels;
- two types of transitions are defined:
 - action transitions $(l, v) \xrightarrow{a} (l', v')$ such that exists an edge $(l \xrightarrow{g, a, r} l') \in E$ where v satisfies g , v' satisfies $v[r]$ and v' satisfies $I(l')$,

- *delay transitions* $(l, v) \xrightarrow{d} (l', v')$ if $\forall d' \in [0, d] \Rightarrow v + d$ satisfies $I(l)$.

Let v_0 denotes the valuation such that $v_0(x) = 0, \forall x \in \mathcal{C}$. If v_0 satisfies the invariant of the initial location l_0 , (l_0, v_0) is called the *initial state* of $T(A)$.

Timed automata are composed into a *network of timed automata* consisting of n timed automata $A_i = (L_i, l_i^0, \mathcal{A}, E_i, I_i), i = 1 \dots n$ over a set of clocks \mathcal{C} . Let $\bar{l} = (l_1, \dots, l_n)$ be a location of the network, then invariants are composed using conjunction $I(\bar{l}) = \bigwedge_{i=0}^n I_i(l_i)$.

Definition 4. Let $A_i = (L_i, l_i^0, \mathcal{A}, C, E_i, I_i), i = 1 \dots n$ be a *network of n timed automata*. Let $\bar{l}_0 = (l_1^0, \dots, l_n^0)$ be the *initial location vector*. Then the semantics is defined as a *transition system* (S, s_0, \rightarrow) , where $S = (L_1 \times \dots \times L_n) \times \mathbb{R}^{\mathcal{C}}$ is the *set of states*, $s_0 = (\bar{l}_0, v_0)$ is the *initial state* and transition relation contains three types of transitions:

- *time flow transitions* $(\bar{l}, v) \xrightarrow{d} (\bar{l}, v + d)$, if $\forall d' \in [0, d]$ holds $v + d' \models \text{Inv}(\bar{l})$;
- *discrete transitions*
 - *synchronized* $((l_1, \dots, l_i, \dots, l_j, \dots, l_n), v) \xrightarrow{\tau} ((l_1, \dots, l'_i, \dots, l'_j, \dots, l_n), v')$ if $\exists i \neq j$, $\exists (l_i \xrightarrow{a^1, g_i, r_i} l'_i) \in E_i, \exists (l_j \xrightarrow{a^2, g_j, r_j} l'_j) \in E_j$, $v \models g_i \wedge g_j, v' \models v[r_i \cup r_j]$ and $v' \models I_i(l'_i) \wedge I_j(l'_j) \wedge \bigwedge_{k \neq i, j} I_k(l_k)$;
 - *asynchronous* $((l_1, \dots, l_i, \dots, l_n), v) \xrightarrow{\tau} ((l_1, l'_i, \dots, l_n), v')$ if $\exists (l_i \xrightarrow{a^1, g_i, r_i} l'_i) \in E_i, v \models g_i, v' \models v[r_i]$ and $v' \models I_i(l'_i) \wedge \bigwedge_{k \neq i} I_k(l_k)$.

There are many tools for designing real-time systems based on the theory of timed automata. For example, KRONOS performs model-checking of TCTL formulas with respect to timed safety automata [24]. The Hybrid Technology tool (HYTECH) is for analysis of embedded systems. It computes the condition under which a linear hybrid system satisfied a temporal requirement. Since times automata are particular hybrid systems they can be verified with this tool [25]. State Graph Manipulator tool (SGM) is for real-time system specification and verification. It uses various sophisticated verification techniques developed in the previous years [26]. The model-checker Uppaal is based on the theory of timed automata as well, however its modeling language offers additional features such as *bounded integer variables* and *urgency*. The query language of Uppaal, used to specify properties to be checked, is a subset of Real Time CTL (computation tree logic) [12], [15], [27]:

- **A[]** *property invariant*, property always holds in all paths;
- **A<>** *property eventually*, property holds in all paths at some moment;
- **E<>** *property possibly*, property eventually holds at some state, at least in one path;
- **E[]** *property potentially always*, property eventually holds from some state, at least in one path;

- **p** \rightarrow **q** *leads to*, whenever p holds eventually q will hold;
- **deadlock true**, if *deadlock* state is reachable;
- **P.state** certain properties hold in the selected state.

IV. UPPAAL MODEL OF THE RADIATION TREATMENT SYSTEM

A work in progress is presented, a simplified version of the radiation treatment system defined in sect. II. Model presented in [21] with timing aspects is extended. Uppaal model consists of the following components:

- **Controller** that, based on its state, a treatment plan and the input from the tracking system, i.e. stereo camera, controls movement of the HexaPOD;
- **HexaPOD** moves according to its physical limitations and following the commands sent from the Controller.
- **HexaPOD Buffer** that models asynchronous communication and latency between the controller and the HexaPOD.
- **Tracker**, in this case an abstraction of a tracking device (e.g., stereo camera), observes tracker placed on the HexaPOD (or patient), calculates position of the tracker, and provides it to the controller. In this model we use predefined inputs and ignore it.

V. GLOBAL DEFINITIONS, VARIABLES AND DESCRIPTION OF COMPLETE SYSTEM

Global definitions and variables are used all over the model. We provide them below.

```
const int X_MAX = x_coord_max;
const int Y_MAX = y_coord_max;
const int Z_MAX = z_coord_max;
const int HP_LATENCY_MAX = hp_latency_max;
const int HP_LATENCY_MIN = hp_latency_min;
const int HP_STEP = hp_step_duration;
```

```
typedef struct {
    int x;
    int y;
    int z;
} POSITION;
```

```
chan move_to; // Controller->HexaPODBuffer
POSITION set_target_pos = {0, 0, 0};
urgent chan get_move; // Buffer->HexaPOD
```

System definition just instantiates all templates and merges them into a complete model.

```
Controller      = Controller_();
HexaPOD         = HexaPOD_();
HexaPODBufferLat = HexaPODBufferLat_();
system HexaPOD, HexaPODBufferLat, Controller;
```

A. HexaPOD

An Uppaal model of the HexaPOD is depicted in Fig. 3. It is modeled as a one point-device with a discrete movement in three - x, y and z directions. We abstract from the acceleration and rotation. Instead of continuous behavior discrete steps on the grid with constant velocity are defined. It allows investigating an impact of the latency and the general design of HexaPOD control. The automaton consists of three locations:

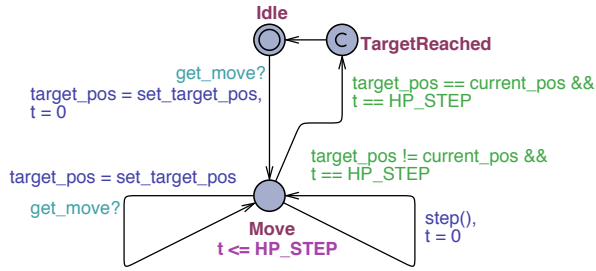


Fig. 3. HexaPOD in Uppaal.

- **Idle:** HexaPOD waits for a command `move_to`. With this action it receives a target, and changes to **Move** location.
- **Move:** HexaPOD stepwise moves towards the target, taking steps in the predefined direction of the predefined length at a constant speed. After each step it checks for a new target, and updates the current one, if necessary. When the target is reached, it changes to **TargetReached** location.
- **TargetReached:** is a committed location (a special type of location, which should be left at the next step), which is used for diagnostic reasons, see sect. III.

Description of the model is defined as follows.

```

clock t;
POSITION target_pos = {0, 0, 0};
POSITION current_pos = {0, 0, 0};
void step() // make step
{
    if (current_pos.x < target_pos.x)
        current_pos.x++;
    else if (current_pos.x > target_pos.x)
        current_pos.x--;
    if (current_pos.y < target_pos.y)
        current_pos.y++;
    else if (current_pos.y > target_pos.y)
        current_pos.y--;
    if (current_pos.z < target_pos.z)
        current_pos.z++;
    else if (current_pos.z > target_pos.z)
        current_pos.z--;
}

```

B. HexaPODBuffer

HexaPODBuffer, depicted in Fig. 4, models asynchronous communication and latency. It consists of the following three locations.

- **Empty** location denotes an empty buffer, it awaits for an input from the Controller, i.e. the `move_to` command, and the target, and then changes to **Latency** location.
- **Latency** location is used to model delays in the system, i.e. after receiving the new target the buffer delays for a while before making it available to the HexaPOD. However, the new target can be provided to the buffer anytime.
- **Ready:** when the buffer is ready, the target can be acquired by the HexaPOD using `get_move` command (action), and location is changed to **Empty**.

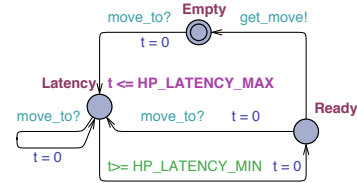


Fig. 4. HexaPODBuffer in Uppaal.

Description of the buffer (just clock) is provided below.

```

clock t;

```

C. Controller

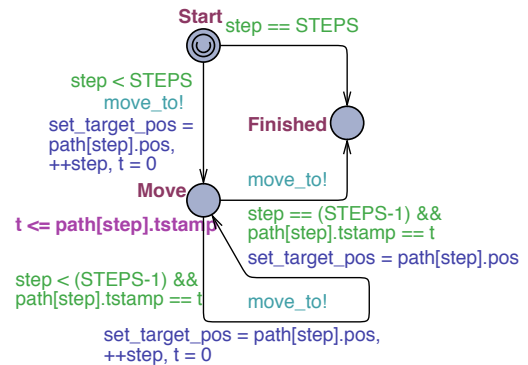


Fig. 5. Controller in Uppaal.

In the current model controller provides control commands to the HexaPOD. It consists of three locations:

- **Start** - start of the treatment program (plan).
- **Move** - the control program is in progress, control inputs provided by an array are sent to the HexaPOD at **the predefined time moment**.
- **Finished** denotes that the control program was completed successfully.

```

const int STEPS = 5;
int step = 0;
clock t;

```

```

typedef struct {
    POSITION pos; // position
    int tstamp; // timestamp
} PATH;

// Test trajectory
const PATH path[STEPS] = {
    {{ 2, 3, 4 }, 0},
    {{ 3, 3, 4 }, 30},
    ...
    {{ 3, 3, 4 }, 10},
    {{ 4, 3, 3 }, 20},
    {{ 3, 3, 3 }, 20}
};

```

D. Simulation and Analysis

Stepwise timed simulations allows to acquire an insight of the model behavior. However, Uppaal allows more, i.e. the conformance of the system to the selected properties can be verified. The following properties are used to analyze the model:

- 1) `E<> Controller.Finished`
property allows to check, if there exists a path that allows for the Controller to reach its final location. This property holds for the model under analysis.
- 2) `A<> Controller.Finished`
allows to check, if the Controller reaches **Finished** location in all evolutions. Verification shows that property holds.
- 3) `E<> Controller.Finished and Controller.step == Controller.STEPS`
there exists such state, that the Controller finishes when all control commands were sent. The property holds for the model as well.
- 4) `E<> Controller.Finished and Controller.step != Controller.STEPS`
checks, if all control steps were performed before reaching the final state of the Controller, i.e. property would hold if there exists at least one state in one path where Controller reaches **Finished** state, but not all control commands were sent. It is formulated in such a way that when the tool returns negative answer, then the system works as expected. The property does not hold. It can be reformulated in a different manner
`A<> Controller.Finished and Controller.step == Controller.STEPS-1`
i.e. we can check, if in all paths eventually the state, where Controller has finished and it has made all steps is reached. It holds.
- 5) `E<> HexaPOD.TargetReached and HexaPOD.current_pos == Controller.path[Controller.STEPS-1]`
there exists such state that HexaPOD reaches the target and its position coincides with the target position set by Controller.
- 6) `E<> HexaPOD.TargetReached and HexaPOD.current_pos != Controller.path[Controller.STEPS-1]`
there exists such state that HexaPOD reaches the target and its position does not coincide with the target position set by Controller. Again, the property is formulated in such a way, that when the tool returns negative answer, then the system is corrected. As expected, the property does not hold.
Again, it can be reformulated in the following manner
`A<> HexaPOD.TargetReached and HexaPOD.current_pos == Controller.path[Controller.STEPS-1].pos`
i.e. we check, if in all paths eventually the state, where HexaPOD has reached target, and it coincides with the last target set by Controller.

Provided properties allow checking different characteristics of the systems and producing diverse diagnostic traces. The traces can be compared to the required trajectories, and the control properties of the HexaPOD as well as the Controller, estimated. More properties can be added. Moreover, traces can be exported and difference between the target and HexaPOD position calculated.

However, as it was already mentioned in [28], [29], it is easy to see that **an average distance between the position of HexaPOD and its target** should be found, and therefore exact durations are needed, and the change of the distance over time. Moreover, more realistic respiratory movement input are necessary. Therefore, hybrid models is required to estimate all durations and time scales.

ACKNOWLEDGMENT

The authors would like to thank UAB Rubedo and in particular Gabrielius Čaplinskas for experiments with HexaPOD couch and comments on its behavior.

VI. CONCLUSIONS AND FUTURE PLANS

A work in progress, the model of the radiation treatment system in Uppaal, is discussed. It is an abstract model, that includes selected elements of the complete system. It allows to obtain some useful characteristics of the system. Moreover, it shows certain limitations of the approach, time scales and corresponding distances (grid) should be chosen to get accurate results. In addition, current system does not allow providing realistic respiratory movement input, it is impossible to calculate average exposure of the healthy tissue.

Our conclusion is that such model is insufficient to answer all the interesting questions, and therefore it should be combined with hybrid model to get more information about the behavior of the system. Our future plans are as follows:

- Extensions of the Uppaal model:
 - model of HexaPOD with acceleration;
 - model of the targeting component;
 - implementation of the different control approaches.
- Continuous model of the HexaPOD, that would allow to build more exact discrete model, or generate discrete paths for timed model.
- Semi-formal control model in OpenModelica [30] (see [29] for the first attempt).
- Combination of the real respiratory movement trajectories and (formal) model to investigate systems adequacy to compensate it.

Moreover, hybrid model results should be used to modify existing models, namely distances and timing. Hybrid and timed simulation results should be compared to validate both models, and different control strategies should be analyzed with models.

REFERENCES

- [1] P. J. Keall, G. S. Mageras, J. M. Balter, R. S. Emery, K. M. Forster, S. B. Jiang, J. M. Kapatoes, D. A. Low, M. J. Murphy, B. R. Murray, C. R. Ramsey, M. B. V. Herk, S. S. Vedam, J. W. Wong, and E. Yorke, "The management of respiratory motion in radiation oncology report of aapm task group 76," *Medical Physics*, vol. 33, no. 10, pp. 3874–3900, 2006. [Online]. Available: <http://link.aip.org/link/MPH/33/3874/1>

- [2] R. I. Berbeco, S. Nishioka, H. Shirato, G. T. Y. Chen, and S. B. Jiang, "Residual motion of lung tumours in gated radiotherapy with external respiratory surrogates," *Physics in Medicine and Biology*, vol. 50, no. 16, pp. 3655–3667, 2005. [Online]. Available: <http://stacks.iop.org/0031-9155/50/i=16/a=001>
- [3] D. Ruan, J. A. Fessler, and J. M. Balter, "Real-time prediction of respiratory motion based on local regression methods," *Physics in Medicine and Biology*, vol. 52, no. 23, pp. 7137–7152, 2007. [Online]. Available: <http://stacks.iop.org/0031-9155/52/i=23/a=024>
- [4] A. Kalet, G. Sandison, H. Wu, and R. Schmitz, "A state-based probabilistic model for tumor respiratory motion prediction," *Physics in Medicine and Biology*, vol. 55, no. 24, pp. 7615–7631, 2010. [Online]. Available: <http://stacks.iop.org/0031-9155/55/i=24/a=015>
- [5] H. Wu, G. C. Sharp, B. Salzberg, D. Kaeli, H. Shirato, and S. B. Jiang, "A finite state model for respiratory motion analysis in image guided radiation therapy," *Physics in Medicine and Biology*, vol. 49, no. 23, p. 5357, 2004. [Online]. Available: <http://stacks.iop.org/0031-9155/49/i=23/a=012>
- [6] R. Milner, *Communication and Concurrency*. Pren.-Hall, 1989.
- [7] C. A. R. Hoare, *Communicating Sequential Processes*. Prent.-Hall, 1985.
- [8] E. Brinksma, T. Krilavičius, and Y. S. Usenko, "Process Algebraic Approach to Hybrid Systems," in *Proc. of 16th IFAC World Congress*, Prague, Jul. 2005, pp. 1–6.
- [9] T. Krilavičius, "Hybrid techniques for hybrid systems," Ph.D. dissertation, Enschede, 2006. [Online]. Available: <http://doc.utwente.nl/57124/>
- [10] D. A. van Beek, K. L. Man, M. A. Reniers, J. E. Rooda, and R. R. H. Schiffelers, "Syntax and Consistent Equation Semantics of Hybrid Chi," *JLAP*, vol. 68, no. 1-2, pp. 129–210, 2006.
- [11] T. Krilavičius and K. Man, *Intelligent Automation and Computer Engineering*. Springer, 2009, ch. Behavioural Hybrid Process Calculus for Modelling and Analysis of Hybrid and Electronic Systems.
- [12] R. Alur and D. Dill, "The theory of timed automata," in *Real-Time: Theory in Practice*, ser. Lecture Notes in Computer Science, J. de Bakker, C. Huizing, W. de Roever, and G. Rozenberg, Eds. Springer Berlin / Heidelberg, 1992, vol. 600, pp. 45–73, 10.1007/BFb0031987. [Online]. Available: <http://dx.doi.org/10.1007/BFb0031987>
- [13] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *TCS*, vol. 138, no. 1, pp. 3–34, 1995.
- [14] G. J. Holzmann, *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, sep 2003. [Online]. Available: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0321228626>
- [15] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on UPPAAL," in *Formal Methods for the Design of Real-Time Systems: 4th Int. School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM-RT 2004*, ser. LNCS, M. Bernardo and F. Corradini, Eds., no. 3185, sep 2004, pp. 200–236. [Online]. Available: <http://www.cs.auc.dk/~adavid/publications/21-tutorial.pdf>
- [16] J. P. Bowen and M. G. Hinchey, "Ten commandments of formal methods," *IEEE COMPUTER*, vol. 28, pp. 56–63, 1994.
- [17] K. Wan, D. Hughes, K. Man, T. Krilavičius, and S. Zou, "Investigation on composition mechanisms for cyber physical systems," vol. 2, no. 1, pp. 30–40, 2010.
- [18] B. Gebremichael, T. T. Krilavičius, and Y. S. Usenko, "A formal model of a car periphery supervision system in UPPAAL," in *Proc. of Workshop on Discrete Event Systems (WODES'04)*, Reims, France, Sep 2004, pp. 433–438.
- [19] K. Man, T. Krilavičius, C. Chen, and H. Leung, "Application of bhv toolset for systems control and mixed-signal design," in *Proc. of the Int. MultiConf. of Engineers and Computer Scientists (IMECS)*, Hong Kong, March 2010.
- [20] T. Krilavičius and V. Miliukas, "Functional modelling and analysis of a distributed truck lifting system," in *The 5th Int. Conf. on Electrical and Control Technologies (ECT 2010)*, Kaunas, Lithuania, 2010, p. 6.
- [21] K. Man, T. Krilavičius, K. Wan, H. D., and K. Lee, "Modeling and analysis of radiation therapy system with respiratory compensation using Uppaal," in *Proc. of the 9th IEEE Int. Symp. on Parallel and Distributed Processing with Application (ISPA 2011)*, May 2011.
- [22] H. Chung, H. Jin, T. Suh, J. Palta, and S. Kim, "Characterization of a commercial add-on couch, HexaPOD™ 6D robotic treatment couchTOP," in *World Congress on Medical Physics and Biomedical Engineering 2006*, ser. IFMBE Proceedings, R. Magjarevic, R. Magjarevic, and J. H. Nagel, Eds. Springer Berlin Heidelberg, 2007, vol. 14, pp. 1945–1947, 10.1007/978-3-540-36841-0_485. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-36841-0_485
- [23] Elekta, "Elekta Synergy® S with HexaPOD™," Online, 2006. [Online]. Available: http://www.elekta.com/assets/Elekta-Oncology/Stereotactic-Radiation-Therapy/case_studies/0601%2003-06%20HexaPOD.pdf
- [24] S. C. Daws, A. Olivero and S. Yovine., "The tool kronos," in *Hybrid Systems III, Verification and Control 1996*, 1996.
- [25] P. T. A. Henzinger and H. Wong-Toi, "Hytech: A model checker for hybrid system," vol. 1, no. 1, pp. 110–122, 1997.
- [26] F. Wang and P.-A. Hsiung, "Efficient and user-friendly verification," vol. 51, no. 1, pp. 61–83, 2002.
- [27] E. Clarke, O. Grumber, and D. Peled, *Model Checking*. MIT, 2001.
- [28] T. Krilavičius and K. Man, "Timed model of the radiation therapy system with respiratory motion compensation," in *Proc. of the 6th Int. Conf. on Electrical and Control Technologies (ECT 2011)*, 2011, p. 6.
- [29] T. Krilavičius, D. Vitkutė-Adžgauskienė, and K. Šidlauskas, "Simulation of the radiation therapy system for respiratory movement compensation," in *Proc. of Mechatronic Systems and Materials (MSM 2011)*, Kaunas, Lithuania, 2011, p. 6.
- [30] OpenModelica System website, "OpenModelica System," 2009. [Online]. Available: <http://www.ida.liu.se/~{}pelab/modelica/OpenModelica.html>

Tomas Krilavičius is an associate professor at Vytautas Magnus university and Senior researcher at Baltic Institute of Advanced Technologies. He has published a number of papers. Tomas is an honorary member of Lithuanian Society of Young Researchers and member of Lithuanian Scientific Society. He is a PC member of DATICS workshops and EURAS conference. Tomas research interests are in the area of formal modeling and analysis of Cyber-physical and Hybrid systems and application of Natural Language Processing for Information retrieval. Tomas holds Ph.D. from University of Twente, The Netherlands.

Kaiyu Wan is a Lecturer at Xi'an Jiaotong Liverpool University. She has published more than 25 academic papers. Kaiyu's research interests include Software Systems (Context-Aware Systems, Cyber Physical System, Service-oriented Architectures and Web service), and Programming Languages (Intensional Programming Language, Language for Agent Communication and Collaboration, Compiler and Development Frameworks). Kaiyu holds a Ph.D. in Computer Science from Concordia University, Canada.

Kevin Lee is a Lecturer at Murdoch University in Perth, Australia. He received his Ph.D. from Lancaster University; was a Research Associate at the University of Manchester and a postdoctoral fellow at the University of Mannheim. He has published more than 40 refereed academic papers in international conferences and journals. His research interests focus on adaptive and autonomic systems in the areas of High-speed Networking, Sensor Networks, Scientific Workflow Processing, Physiological Computing and Peer-to-Peer networks.

Ka Lok Man holds a Dr. Eng. degree in Electronic Engineering from Politecnico di Torino, Italy; and a Ph.D. degree in Computer Science from Technische Universiteit Eindhoven, The Netherlands. Currently, he is a Senior Lecturer with the Department of Computer Science and Software Engineering, Xian Jiaotong-Liverpool University, China and a research and engineering consultant for Solari, Hong Kong. His research interests include logic synthesis, simulation, formal verification and low power design methodologies for integrated circuits and systems, formalization of SystemC and SystemVerilog design, formal methods, process algebras, wireless sensor networks, specification and analysis of distributed, real-time, hybrid systems, embedded systems and physical cyber systems. On the above-mentioned topics, he has authored or co-authored more than 100 refereed publications.