DOI: 10.15514/ISPRAS-2022-34(6)-11

# Modelling Interrelationship between Diseases with Communicating Stream X-Machines

*D. Jayatilake, ORCID: 0000-0002-3377-7129 <dilshan.jayatilake@uwe.ac.uk>*
*K. Phung, ORCID: 0000-0002-9341-2033 <khoa.phung@uwe.ac.uk>*
*E. Ogunshile, ORCID: 0000-0002-6276-188X <emmanuel.ogunshile@uwe.ac.uk>*
*M. Aydin, ORCID: 0000-0002-4890-5648 <mehmet.aydin@uwe.ac.uk>*

*University of the West of England*
*Coldharbour ln, Bristol, BS16 1QY, UK*

**Abstract.** The world is moving towards alternative medicine and behavioural alteration for treating, managing, and preventing chronical diseases. In the last few decades, diagrammatical models have been extensively used to describe and understand the behaviour of biological organisms (biological agents) due to their simplicity and comprehensiveness. However, these models can only offer a static picture of the corresponding biological systems with limited scalability. As a result, there is an increasing demand to integrate formalism into more dynamic forms that can be more scalable and can capture complex time-dependent processes. In this paper, we introduce a generic disease model called Communicating Stream X-Machine Disease Model (CSXMDM), which has been developed based on X-Machine and Communicating X-Machine theories. We conducted an experiment on modelling an actual disease using a case study of Type II Diabetes. The results of the experiment demonstrate that the proposed CSXMDM is capable of modelling chronic diseases.

**Keywords.** Communicating Stream X-Machine; Stream X-Machine; Type II Diabetes; modelling; formal method; cardiovascular disease

**For citation:** Jayatilake D., Phung K., Ogunshile E., Aydin M. Modelling Interrelationship between Diseases with Communicating Stream X-Machines. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 6, 2022. pp. 147-164. DOI: 10.15514/ISPRAS-2022-34(6)-11

## Моделирование взаимосвязи между заболеваниями с помощью взаимодействующих потоковых Х-машин

*Д. Джаятилаке, ORCID: 0000-0002-3377-7129 <dilshan.jayatilake@uwe.ac.uk>*
*К. Фунг, ORCID: 0000-0002-9341-2033 <khoa.phung@uwe.ac.uk>*
*Э. Огуншиле, ORCID: 0000-0002-6276-188X <emmanuel.ogunshile@uwe.ac.uk>*
*М. Айдин, ORCID: 0000-0002-4890-5648 <mehmet.aydin@uwe.ac.uk>*

*Университет Западной Англии,*
*Великобритания, BS16 1QY, Бристоль, Колдхарбор-лейн*

**Аннотация.** Мир движется к альтернативной медицине и изменению методов лечения, контроля и профилактики хронических заболеваний. В последние несколько десятилетий диаграммные модели широко использовались для описания и понимания поведения биологических организмов (биологических агентов) благодаря их простоте и полноте. Однако эти модели могут предложить только статическую картину соответствующих биологических систем с ограниченной масштабируемостью. В результате растет спрос на интеграцию формализма в более динамичные формы, которые могут быть более масштабируемыми и могут охватывать сложные процессы, зависящие от времени. В этой статье мы представляем общую модель на основе теорий Х-машин и взаимодействующих Х-машин. Мы провели эксперимент по моделированию реального заболевания на примере диабета II типа. Результаты





эксперимента демонстрируют, что предложенный метод способен моделировать хронические заболевания.

**Ключевые слова:** взаимодействующие потоковые Х-машины; потоковая Х-машина; диабет второго типа; моделирование; формальный метод; сердечно-сосудистые заболевания

**Для цитирования:** Джаятилаке Д., Фунг К., Огуншиле Э., Айдын М. Моделирование взаимосвязи между заболеваниями с помощью взаимодействующих потоковых Х-машин. Труды ИСП РАН, том 34, вып. 6, 2022 г., стр. 147-164. DOI: 10.15514/ISPRAS-2022-34(6)-11

## *1. Introduction*

Despite the development in medical domain still the chronical disease (e.g., osteoporosis (OP), gout, rheumatoid arthritis (RA), type 2 diabetes (T2D), Alzheimer's disease (AD)) prevention, mitigation and managing patients who are suffering from chronicle diseases has limited to a single dimension treatment pattern. Still there is a very limited understanding about the etiology and the mechanism of the diseases [1, 2]. It is important to understand the mechanism (behaviour of the disease). Therefore, precise modelling of the diseases will improve treating patients with chronicle diseases and in prevention and management. On the other hand, having a better understanding about the mechanisms of the disease will help in directing the launches of proper clinical trials, target for effective lifestyle interventions and pharmacological innovations.

Human body consists of closely connected subsistence (example: cardiovascular system, digestive system etc.;). Therefore, the human body can be identified as a complex system where continuant subsistence work together in order to keep a human alive [3, 4]. As in any other complex system, there are some undesired states where the human body can reach. in our research we consider diseases as undesired states (similar to the error states in complex systems) [3]. Due to the close coupling among those systems and the complex relationships has made it challenging to precisely model the human body. Because of the complex interrelationships among the subsystems of the body, we believe that diseases mimic the same.

Shaikh F. Hossain et al. [5], has discussed the inter relation between diseases based on the biomarkers. Based on 432 biomarkers mapping in to 18 diseases, authors have been able to demonstrate the chemical and the alliance between the disease. Furthermore, literature provide evidence of the relationship between multiple chronic diseases [6], and effects on the human behaviour. These further discuss the importance of having clear and precise understanding for treating patients with such conditions.

With the evolving research and development in the fields of metabolomics, proteomics, and nutrigenomics are shifting the perspective of nutrition and diet management to be considered as medicine and concept of nutrition therapeutics [7] are actively used in experimental patient management [8]. Due to the demand of utmost accuracy and precision in this domain, to successfully implement in practice has been a challenge.

Literature provides evidence of employing formal methods in modelling critical systems where they have demanded very high level of accuracy [9-11]. In our previous research formal methods has been successfully used modelling diseases [3].

Furthermore, we discuss the significance of the accuracy modelling diseases and the importance of having the ability to communicate between diseases to demonstrate the impact of one disease has towards the other disease(s). During this research we have decided to use well known formal method namely, Communicating Stream X-Machines (CSXM), an extension of Stream X-Machine (SXM) to model diseases and the relationship among them.

This paper is structured as follows. Section 2 discusses the related work. Section 3 and Section 4 outline the proposed disease model with the employment of Stream X-Machine and Communicating Stream X-Machine, respectively. Section 5 demonstrates the experiment and its results. Section 6 consists of the discussions about the experimental results and the proposed disease model. Section 7 concludes the paper and outlines future research directions.

## *2. Related Work*

### 2.1. Finite State Machine, X-Machine, and Stream X-Machine

X-Machine (XM) [12] is an extension of the finite state machine (FSM), where the FSM is empowered with a data structure and a processing function(s). As explained, XM is very similar to an FSM with the difference of having a data sent X (type of the machine), where finite set of processing functions Φ operates on the X (φ: X → X | φ ∈ Φ) and inbuilt memory (M) where thy transitions can alter the memory. In the FSM diagram each transition (arch) is ladled with a processing function of φ (φ | φ ∈ Φ). Because of the generalisation of the type X, X-Machine has proven that it is capability of modelling any type of system accurately and also in a very generic manner where extension and/or modification is possible [11-14, 16].

Another class of X-Machine has been introduced to specify and model software namely Stream X-Machines (SXM), which has the ability to define and validate data types and the functionalities of a software systems [17]. The definition of SXM is as follows.

$$\mathbf{Z = (\Sigma, \Gamma, Q, M, \Phi, F, q_0, m_0)}$$

where:

- Σ = input alphabet;
- Γ = output alphabet;
- Q = finite set of states;
- M = memory (possibly infinite);
- Φ = the type of the machine X, where a set of partial functions $\varphi(\varphi|\varphi \in \Phi)$ which transforms input and a memory to an output and a possibly to a different memory state: φ: Σ × M → Γ × M;
- F = Next state partial function which drives the machine state: F: Q × Φ → Q;
- $q_0$, $m_0$ = initial state and the initial memory of the machine respectively.

For the above illustrated SXM, the associated FSM is **Az = (Φ, Q, F, $q_0$)**. This Az is also known as associated finite state automaton (FA). Fig 1 illustrates the state-transition diagram of a three-state Stream X-Machine.

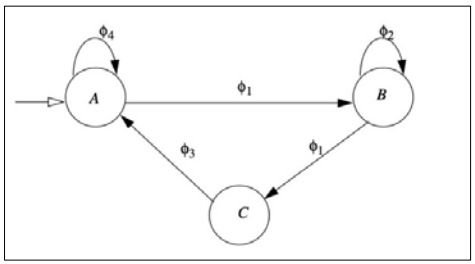


*Fig. 1. A three-state Stream X-Machine*

### 2.2. Communicating Stream X-Machine (CSXM)

There are number of different approaches can be found for CSXM in the literature [18-20]. In this research, we discuss the standard approach and an alternative approach which leads to the development of SXM – Communicating Component (SXMCM).

#### 2.2.1 Communicating Stream X-Machine standard approach

Communicating Stream X-Machine which consists of n number of constituent components can be identified [18, 21] as a tuple of: $((XM_i)_{i=1..n}, CM, C_0)$ where:

- $XM_i$ is the $i^{th}$ X-Machine of the system;





- $CM$ is a n×n matrix, namely the Communication Matrix;
- $C_0$ is the initial communication matrix.

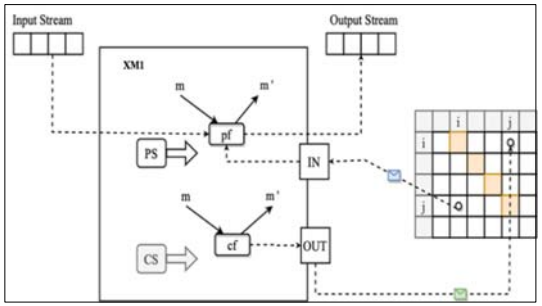


*Fig 2. CSXM illustration with Communication Matrix*

In this approach, there is a difference from the usual standalone SXM definition because the $XM_i$ has defined IN and OUT ports defined for communication (**Ошибка! Источник ссылки не найден.**) [21]. As illustrated in **Ошибка! Источник ссылки не найден.**, both IN and OUT ports are connected to a communication matrix (CM). CM facilitates the communication between the SXMs. CM cells holds me messages from the SXMs. For example, cell (i, j) has the message that was sent from $SXM_i$ to $SXM_j$. The type of the message can be any type which is defined in the memory. λ stands for empty or no message. Communication is initiated only from a communicating state, which accepts empty symbol ε as an input and produces ε as the output without effecting the memory. Communicating function only either reads an element from the associated CM and writes in the IN port and assign the value of λ to the cell in which the function read from or writes an element from the OUT port to CM if the cell has no message (cell has the value of λ) [18, 21]: **cf(ε, m, in, out, c) = (ε, m, in', out', c')** where m ∈ M; in, in' ∈ IN; out, out' ∈ OUT; and c, c' ∈ CM.

If the communication function is not applicable, it waits. The processing function only affects the ports (IN & OUT) but do not affect the communication matrix) [18, 21]: **pf(σ, m, in, out) = (γ, m', in', out')** where: σ ∈ Σ; m, m' ∈ M; in, in' ∈ IN; out, out' ∈ OUT; and γ ∈ Γ.

#### 2.2.2 Communicating Stream X-Machine alternative approach

Standard CSXM suffers from inability to conceive as an independent component. Due to this limitation it has to be started from the beginning in order to add a new component to the arrangement. Furthermore, this limits the ability of the components in this arrangement to act as a standalone SXM or be a part in other arrangements.

Petros Kefalas et al. [21] proposed an X-Machine type (MT) without the initial state or memory as an alternative approach to overcome the previously identified issues. MT is defined as follows:

$$\mathbf{MT = (\Sigma, \Gamma, Q, M, \Phi, F).}$$

An X-Machine is constructed through application of operators:

$$\mathbf{OP\ inst:\ MT_i \times (q_{0i}, m_{0i}) \rightarrow M_i, \forall\ q_{0i} \in Q, m_{0i} \in M.}$$

That creates the instance of MT: **M = MT OP inst ($q_0$, $m_0$).**

Communicating X-Machine Component is defined as:

$$\mathbf{XMC_i = (\Sigma_i, \Gamma_i, Q_i, M_i, \Phi_i, F_i)\ OP\ inst\ (q_{0i}, m_{0i})\ OP\ comm\ (IS_i, OS_i, \Phi_i, S_i, \Phi_{OSi})}$$

where:

- $IS_i$ is a tuple with *n* input streams, which contain the source of the message where it has been generated (CSXM which send the message) (in i is the standard input source of $CSXM_i$): $IS_i$ = ($is_1$, $is_2$, ...$is_i$, ..., $is_n$), and $is_j$ = ε (if no communication is required) or $is_j \subseteq \Sigma_j$;
- $OS_i$ is a tuple defined in correspondence with *n* output streams, which consists of the destination

of the *n* message that used in generating the message. destination of $CSXM_i$): $OS_i = (os_1, os_2, ..., os_i, ..., os_n)$, and $os_j = \varepsilon$ (if no communication is required) or $os_j \subseteq \Sigma_j$;

- $\Phi IS_i$ is an association of function $\varphi_i \in \Phi_i$ and the input stream $IS_i$, $\Phi IS_i$: $\varphi_i \leftrightarrow IS_i$;
- $\Phi OS_i$ is an association of function $\varphi_i \in \Phi_i$ and the output stream OSi, $\Phi OS_i$: $\varphi_i \leftrightarrow OS_i$.

Applying the operator $\mathbf{OP_{comm}: XM_i \times (IS_i, OS_i, \Phi_{ISi}, \Phi_{OSi}) \rightarrow CXMC_i}$ has a result a Communicating X-machine component $CXMC_i$ as a tuple:

$$\mathbf{SXMC_i = (\Sigma_i, \Gamma_i, Q_i, M_i, \Phi_{Ci}, F_i, q_0, m_0, IS_i, OS_i)}$$

where:

- $\mathbf{\Phi_{Ci}}$ is a set of partial functions that read from standard input or any other input stream and write to standard output or any other output stream. Such set will consist of 4 different sets of function: $\mathbf{\Phi_{Ci} = SISO_i \cup SIOS_i \cup ISSO_i \cup ISOS_i}$, where:
  - $SISO_i$ is a set of functions φ which reads and writes to standard input ($is_i$) and standard output ($os_i$) streams respectively. $SISO_i = \{(is_i, m) \rightarrow (os_i, m) \mid \varphi i = (\sigma, m) \rightarrow (\gamma, m) \in \Phi_i \wedge \varphi_i \notin dom(IS_i) \wedge \varphi_i \notin dom(OS_i)\}$;
  - $SIOS_i$ – a set of functions φ which read and writes to standard input ($is_i$) and $j^{th}$ output ($os_j$) streams respectively. $SIOS_i = \{(is_i, m) \rightarrow (os_j, m) \mid \varphi_i = (\sigma, m) \rightarrow (\gamma, m) \in \Phi_i \wedge \varphi_i \notin dom(IS_i) \wedge (\varphi_i \rightarrow os_j) \in OS_i\}$;
  - $ISSO_i$ is a set of functions φ which reads and writes to $j^{th}$ input ($is_j$) and standard output ($os_i$) streams. $ISSO_i = \{(is_j, m) \rightarrow (os_i, m) \mid \varphi_i = (\sigma, m) \rightarrow (\gamma, m) \in \Phi_i \wedge (\varphi_i \rightarrow is_j) \in IS_i \wedge \varphi_i \notin dom(OS_i)\}$;
  - $ISOS_i$ is a set of functions φ which read and writes to $j^{th}$ input ($is_j$) and $k^{th}$ output ($os_k$) streams. $ISOS_i = \{(is_j, m) \rightarrow (os_k, m) \mid \varphi_i = (\sigma, m) \rightarrow (\gamma, m) \in \Phi_i \wedge (\varphi_i \rightarrow is_j) \in IS_i \wedge (\varphi_i \rightarrow os_k) \in OS_i\}$.

Communicating X-Machine is defined as a tuple of *n* XMC as:

$$\mathbf{CXM = (XMC1, XMC2, ..., XMCn)} \text{ with}$$

- $\Sigma_1 \cup \Sigma_2 \cup ... \cup \Sigma_n = (os_{11} \cup os_{12} \cup ... \cup os_{1n}) \cup ... \cup (os_{n1} \cup os_{n2} \cup ... \cup os_{nn})$;
- $\Gamma_1 \cup \Gamma_2 \cup ... \cup \Gamma_n = (is_{11} \cup is_{12} \cup ... \cup is_{1n}) \cup ... \cup (is_{n1} \cup is_{n2} \cup ... \cup is_{nn})$.

In this approach, the CM has been replaced by several input streams associated with each Stream X-Machine component. Through replacement of the communication state and the functions with functions that belongs to type Φ and providing the ability to read and write to different input and output streams has improve the usability of the models (fig. 3) [21].

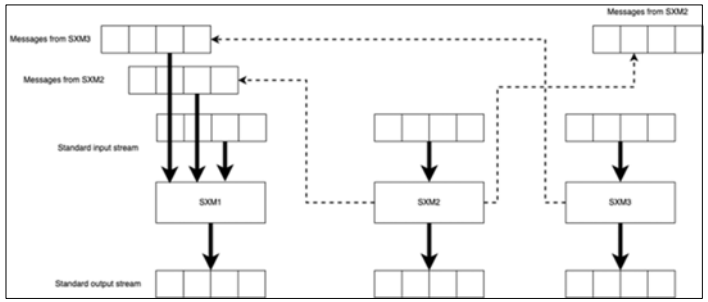


*Fig. 3. Three Communicating Stream X-Machines*

## 2.3 Disease modelling with formal methods

Biological systems such as human body consists of tightly connected components (organs) that change their actions and behaviours over the time and their interactions with exposure to external factors (e.g., nutrients, viruses, bacteria). In addressing such challenges, literature reveals that, there are few formal approaches has been occupied namely, Boolean Networks (BN) and its extensions





(i.e. Qualitative Networks (QN), Gene Regulatory Networks (GRN) [22-24], Petri Nets (PR), Cellular Automata (CA), population P systems (PPS), etc.

*Table 1. Review of formal methods*

| Method | Strengths | Weaknesses | Example |
|---|---|---|---|
| BN/QN [15, 16, 18] | Correctly defined binary states. | Due to lack of state space in BN (Active and Inactive), harder to model complex state models such as Biological Systems.<br>Even though, QN discrete variable values, and the dependencies of those values are expressed with algebraic functions instead of Boolean functions, complex relations mapping is not possible. | BN/QC has been employed in Schematic view of signal transduction in the pancreatic cancer model [28]. |
| PR [29, 30] | Has the capability of comprehensive modelling and analysing facility for distributed and concurrent systems. | The ability of providing the concurrency the complexity of the model increases significantly.<br>PR suffers from the inability to test in unbounded places. | PR has been used to model enzymatic reactions in Metabolic pathways [31]. |
| CA [25] | Capable of modelling interactive components.<br>CA is a powerful way of defining agent-based system due to the simplicity. | CA presents challenges in modelling non-trivial systems due to lack of data representation.<br>CA suffers from state and input symbol explosion | CA has been occupied in modelling virus behaviour (e.g. Human Immunodeficiency Virus (HIV)) [32]. |
| PPS [26] | PPS provides a robust mechanism to introduce new nodes, remove nodes, and change the behaviour of the defined nodes, which helps to map the disease behaviour. | PPS lacks the ability to represent internal states and individual behaviour of the nodes. | PPS has been employed in modelling species behaviour, especially when it eloper is introduced to the system (e.g. behaviour on school of fish upon a predator) [33]. |

As evaluated in Table 1, current formal specification models have not been able to comprehensively model biological systems. Even we consider the human body as a biological system, each disease has its own behaviour and attributes which are interconnected yet different. Considering the BN and QN would have the capability of modelling the binary state of biological systems yet suffers from lack of state space [15, 16, 18]. When considering diseases, there are multiple states for most of the diseases. For instance, diabetes has few inner states such as Pre-metabolic syndrome, metabolic syndrome pre-diabetic and Type II Diabetes. Furthermore, this will not facilitate the interconnectivity with expected efficiency and the accuracy. PR would mostly address the issues that has been identified with BN and QN. PR provides a comprehensive modelling and analysing facilities for distributed agents (systems), which would be ideal for modelling diseases; hence, the anonymity and the connectivity (in terms of having relationships between diseases) at the same time

[26, 27]. However, the concurrency introduces significant and unmanaged complexity to the system. Diseases act in concurrently in nature. Therefore, this would bring unexpected and unrealistic complexity to the models, which hinders the demanded accuracy. CA addresses the above discussed complexity and has been proven to be used in agent-based systems [25]. However, CA suffers from data representation, whereas disease models should encapsulate and persist their own data for monitoring disease progression. One other aspect of disease modelling is that diseases are expected to be infected and cured (excluding incurable diseases). The models should be easily introduced and removed from the system. PPR provides the capability of adding, deleting, and changing nodes [26]. However, PPR fails to introduce the initial state and modelling individual behaviour. Therefore, it can be inferred that the reviewed formal methods in Table 1 have not provided a comprehensive mechanism to model diseases and their behaviours completely with the expected accuracy.

There is an evidence of research that they have used formal methods in order to model biological systems, in specific they have employed Communicating X-Machines in order to model the bio-systems where it has used the ability of representing the interrelationship and the communication aspects of the biological systems [26], while preserving the ability of data persistence with flexibility. To the best of our knowledge this is the first attempt that formal verification methods has been occupied in order to model a chronic disease and model the inter-relationship among diseases.

## 2.4 Strengths and weaknesses of existing models

Literature provides evidence of occupying formal methods to model biological systems. However, most of these approaches have not been able to provide the much needed 100% accuracy level in modelling such systems. Hence the medical domain is one of the most critical domains which the cost of a mistake could be a life [15, 27]. In this paper, we have considered few approaches and their strengths and weaknesses (table 1).

As illustrated above, mathematical models present significant challenges in modelling diseases due to the highly complex interrelationship(s) among the large state space and disease models themselves with other disease models. Due to this highly complex and heavily interconnected model behaviour, most of the mathematical models are being challenged, and/or the models become unmanageably complex [15].

Due to the criticality of the domain in nature, to the best of the authors' knowledge, it is expected that the formal modelling approaches would bring more efficiency and accuracy for disease modelling in general, especially X-Machine models would be more realistic due to their capability in modelling critical systems [34], where interrelationships among the models can be more accurately handled. Therefore, in this research we have selected X-Machines in general to model diseases, particularly Stream X-Machine is used to model Type II Diabetes to demonstrate the capability of modelling diseases.

## 2.5 State of the art in biological system modelling and gap analysis

Literature provides many evidence of employing mathematical models to precisely understand the disease behavior [15, 16, 18, 25, 26, 28, 29, 30, 35-38]. These have been always focused on one disease and also disease as a whole and its impact in the perspective of spread, implication the medical sector and/or holistic implication on the society. In our previous publication we demonstrated how formal speciation can be employed in modelling diseases. To the best of our knowledge this the first attempt that, employing formal methods in general especially classes of X-Machine in modelling the behaviour of diseases with representing its impact on the other diseases.

## *3. Stream X-Machine Disease Model (SXMDM)*

We have introduced the SXMDM with the capability of adopting to any diseases, which gives the capability of mimicking the disease behaviour. SXMDM proposed in our previous research has the capability of:





- Describe diseases and their stages (Demonstration/research purposes for the medical professionals).
- Simulate the progression of the diseases (Positive and negative).
- Determine the stages of the diseases at the time of the patient is being diagnosed.
- Determine the stages with the change of the symptoms.
- Understand the progression of the diseases.
- Monitoring the progression of the patient with past information (Symptoms and data).

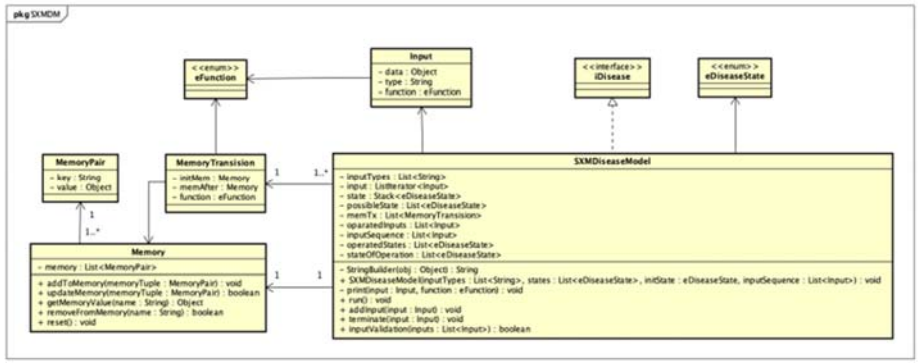


*Fig 4. Class diagram of SXMDM*

The class diagram of the proposed SXM disease model is illustrated in fig 4.

- XMachineDisease: this class contains the generic X-Machine model which is implemented iDiseaseType interface.
- iDiseaseType: this interface provides the necessary functionalities to describe the Type II Diabetic medical condition.
- MemoryPair: this class provides the unit model which holds the memory units.
- eFunction: this special class of constant data (enum) class provides all the processing functions that involves in the defining disease type SXM.
- MemoryTransition: this class contains the memory transition based on the processing function (eFunction).
- Input: this class provides the input unit which holds in the input sequence of the SXM.
- eDiseaseState: this special class of constant data (enum) provides all the states of the disease.

## *4. Communicating Stream X-Machine Disease Model (CSXMDM)*

CSXMDM is an extension of SXMDM proposed SXDM in our previous research [3]. During our last research we have discussed that, similar to the organs (subsystems) of the body work together for the purpose of wellbeing of the human, disease has the similar characteristic of having inter-relationship among them.

New extended CSXMDM has address the about shortcoming of the SXMDM and given the capability of communicating with other disease models. The CSXMDM has been designed as a pluggable component to SXMDM. In order to facilitate preciously defined SXMDM has been slightly modified. CSXMDM is focusing on modelling the diseases and their inter-relationship(s). The CSXMDM we proposed takes a state-based communication approach extended from modular approach. Instead of considering the CSXM as a whole, in the proposed CSXMDM has communicates among the other models with preserving the communication-initiated state, during and after communication.

### 4.1 Changes in SXMDM

We have changed the SXMDM discussed in Section 3 to facilitate the communication. Hence, we are more interested in the state level communication, sates have given the ability to define its readiness to communicate at the state level (fig. 5).

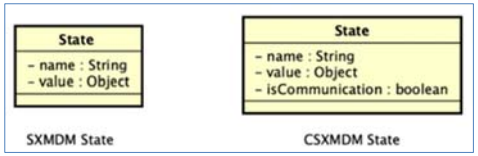


*Fig. 5. Implementation of State in SXMDM & CSXMDM*

### 4.2 SXM – Communicating Component (SXMCM)

In order to support Communications between the SXMDMs, we are proposing SXMCM. The implementation of the SXMCM is illustrated below (fig. 6).

- **CommunicatorBase**: This class provides the basic infrastructure to establish communication between the SXMDMs. This class acts as the common mediator of the communication.
- **Communicator**: This abstract class provides interface for establishing the communication in the SXMDMs (provide that the SXMDM is using the new version of State discussed previously).
- **Message**: This provides the unit model for messages.
- **MessageUnit**: This class facilitates the message buffer for the Communicator.

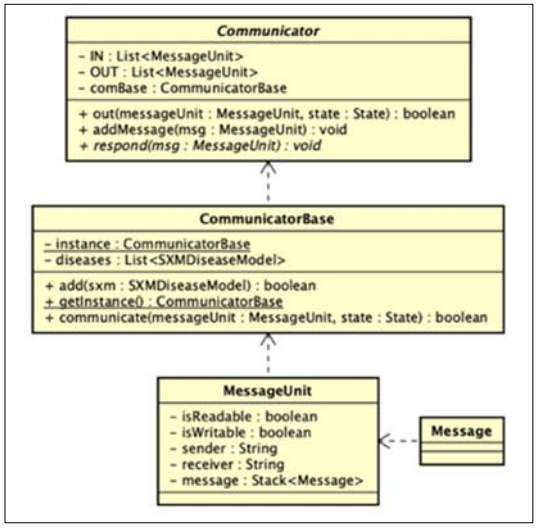


*Fig. 6. Implementation of SXMCM*

### 4.3 Discussion of implementation

SXMCM consists of two main packages. *com.communication.communicationCmp* consists of basic implementation of the communication plugin. This consists of the *CommunicationBase* and *Communicator*. *com.communication.util* provides the utilities for *Communicator* and *CommunicationBase*. This consists of the *Message* and *MessageUnit*.





## *5. Case Study*

We have employed the same case study from our last research with the addition of cardiovascular disease (CVD) model (fig. 7).

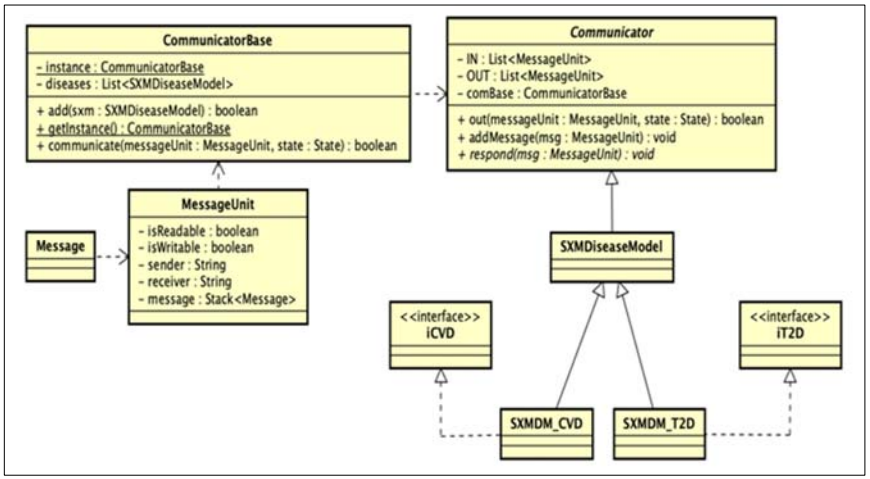


*Fig. 7. Implementation of the Type II Diabetes disease model with CVD rick model*

### 5.1 Type II Diabetes Model

To demonstrate the validity of the Stream X-Machine Disease Model (SXMDM), we consider the implementation of Type II Diabetes Model (TTDM). To test the correctness of the TTDM, we have to adopt the TTDM to SXDM. In our proposed TTDM, we have assumed that the symptom collection is done elsewhere, and the data (Symptom's information) is fed into the model.

### 5.2 Type II Diabetes and Symptoms Evaluation

There are clearly defined four stages of diabetes namely, pre metabolic syndrome, metabolic syndrome, pre diabetic and type II diabetic. Depending on the time of the diagnosis a patient can be in any of the above mentioned four stages. With the time, a patient can progress his/her diabetic stage positively or negatively.

The initial state of the disease (diabetic) is defined by the number of symptoms that the patient is presenting at the time of the diagnosis [13]. The symptom evaluation criteria to define the stage of the diabetic is as follows.

If a patient is presenting with:

- A waist circumference off:
  - 94 centimetres and above for European men or 90 centimetres an above for South Asian men.
  - 80 centimetres or more for South Asian woman.
- High levels of triglyceride in the blood.
- Low levels of HDL.
- Constant levels of high blood pressure 140 / 90mmHg
- Low levels of insulin response (inability to control blood sugar levels)
- A tendency to develop irritation and swelling off the body tissues in general inflammation.

*Table 2. Evaluation criteria*

| Number of Symptoms | Stage of Diabetes |
|---|---|
| 0 | Normal |
| 1-2 | Pre-Metabolic Syndrome |
| 2-3 | Metabolic Syndrome |
| 4-5 | Pre-Diabetes |
| 5 or more | Type II Diabetes |

The Type II Diabetic Stage Identification can be described as follows: The TTDM is in the Start State, waiting to process patient's symptoms data. When the patient information is fed in, to start the evaluation process. Then the machine will validate the inputs (Symptom information) and process the data for Type II Diabetic stage evaluation. Assuming that the input data is valid, machine starts the evaluation process. Once the evaluation is completed the machine will determine the Diabatic Stage of the patient. Then machine will wait until recheck occurs to re-evaluate the patient. When the re-evaluation is called, the machine will follow the same process as before (fig. 8).

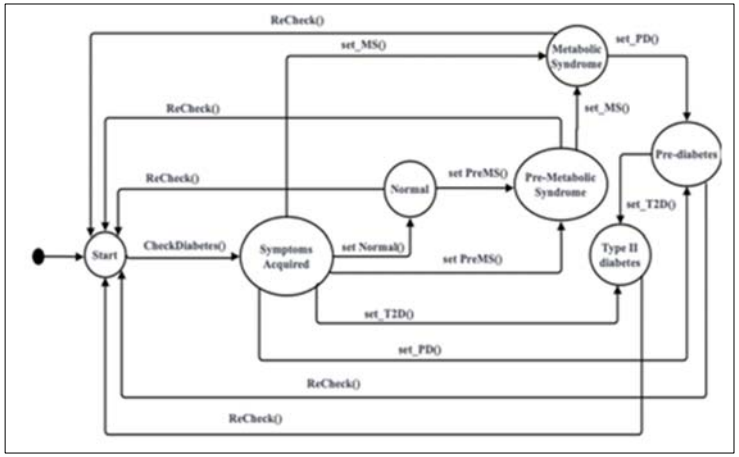


*Fig. 8. Type II Diabetes state-transition diagram*

We consider first stage of the disease as Start and then leads to the state where the symptoms are acquired and validated (Symptoms acquired). Depending on the symptoms that the particular patient presented at the time of the diagnosis, TTDM defines the stage of the disease. Then the TTDM will wait at the particular level till, recall method is called to submit new (updated symptoms), and repeats the same process.

With the above description, SXMDM can be modelled as a SXM with 6 states. The states are as follows:

- Normal: TTDM waits for the recheck data after acquiring the diabetes state.
- Pre-Metabolic Syndrome: TTDM waits for the recheck data after acquiring the diabetes state.
- Metabolic Syndrome: TTDM waits for the recheck data after acquiring the diabetes state.
- Prediabetes
- Type II Diabetes: TTDM waits for the recheck data after acquiring the diabetes state.
- Symptoms Acquired: TTDM waits for identifying the diabetes stage.
- Start (initial state): TTDM waits for the patient data.





The memory of the TTDM consists of one element with accordance to our implementation.

- Diabetes: a data object of the disease which contains all the necessary to evaluate a patient's diabetic stage.

In the TTDM, we have identified that there are seven functions but drives the model and one internal function.

- set_PreMS(Diabetes diabetes);
- set_MS(Diabetes diabetes);
- set_PD(Diabetes diabetes);
- set_Normal(Diabetes diabetes);
- set_T2D (Diabetes diabetes);
- checkDiabetes(Diabetes diabetes);
- getCurrentState();
- reCheck(Diabetes diabetes).

## 5.3 Cardiovascular Disease Risk Model (CVDRM)

CVD has a clear relationship with Type II Diabetes (T2D) [40]. The current model of the CVDRM is designed only with the intention of demonstrating the CSXMDM. The model we have developed for the CVD is capable of modelling the level of risk with relevant to T2D. Currant CDVR consists of four risk states and two operational states including start state namely *Start, CheckCVD, Normal, LowRisk, ModerateRisk, HighRisk*. The model consists of six processing functions that drives through the above states namely, *SET_NORMAL (CVD cvd), CHECKCVD (CVD cvd), SET_LOW (CVD cvd), SET_MODERATE (CVD cvd), SET_HIGH (CVD cvd), RECHECK (CVD cvd).*

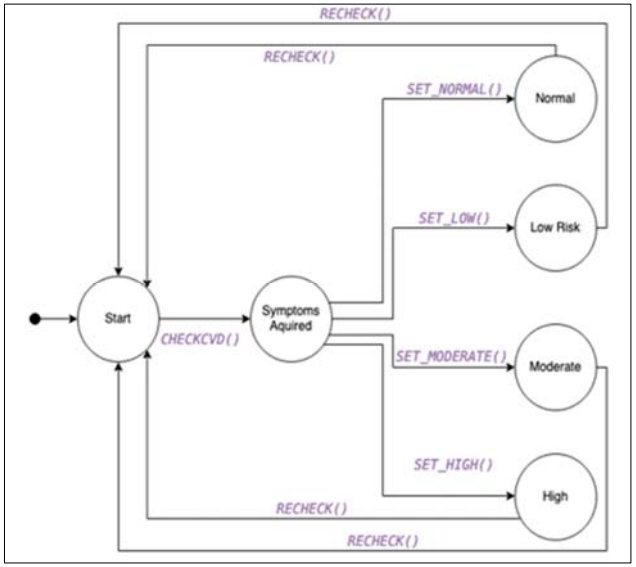


*Fig. 9. CVD state-transition diagram*

For the demonstration purposes, similar to the above discussed TTDM, CVDRM will evaluates the level risk according to the below mentioned criteria. In the context of this research, we consider if a patient is diagnosed with high levels of HDL, high levels of Triglycerides, higher degree of waist circumference and had been diagnosed with any stage of diabetes will be considered as a symptom. In the context of this research, we assume that if a patient is presenting no symptoms, has no risk

(implied by *Normal*), with one symptom has a low risk, with two symptoms has a moderate risk, with three risks will identified as high risk (fig. 9).

### 5.4 Demonstration of the TTDM

For demonstration purposes let us assume that a patient is presenting with:

- Level of HDL: Low;
- Level of Triglycerides: Low;
- Degree of waist circumference: Low;
- Stage of diabetes: Non (Normal) / Not diagnosed yet..

While diagnosing towards the patients' CVD check-up. CVDRM will be in its *Start* state. When the symptoms are given, CVDRM invokes *CHECKCVD(CVD cvd)* processing function which will update the memory with the newly received CVD object and proceeds to evaluate the number of symptoms with against the defined evaluation criteria. Based on the evaluation the input sequence will be amended with the relevant processing function to define the next state. In the given scenario the CVDRM will settles in ***Normal*** state hence there are no positive symptoms.

Thereafter, we will assume that the same patient is being diagnosed with high levels of triglyceride in the blood, low levels of HDL, consistent high blood pressure, a waist circumference in normal range and no complaints about body inflammation and high level of insulin response while being checked for diabetes.

By this time, the TTDM initially will be in the *Start* state. When the symptoms are given, TTDM invokes *checkDiabetes(Diabetes diabetes)* processing function, which updates the memory of Diabetes with the data object. Then the TTDM evaluates the Diabetes data object and analyse the number of symptoms present, against the evaluation criteria. Then amends the input sequence with the relevant processing function. In this scenario, the patient is presenting with four positive symptoms and two negative symptoms.

Therefore, the TTDM will evaluates the symptoms and decides the next state as ***Metabolic Syndrome***. In this scenario any stage of Type II diabetes as a communication state, which means in this context dragonising with Metabolic Syndrome improves the risk of CVD.

Therefore, TTDM will be initiating the communication. TTDM will generate the message and add it to its OUT communication-buffer. This will trigger the communication component. The communication component evaluates the message and checks for the intended receives of the message (in this instance it is the CVDRM). Then the risk factors are evaluated. Based on the received message through the communication buffer, the CVDRM will alter its input stream accordingly. In this scenario this will add one risk factor where the risk state will move from ***Normal*** to ***Low Risk***.

Furthermore, the CVDRM has given a second dataset through *RECHECK(CVD cvd)* with all positive symptoms. This will drive the model to ***High-Risk*** state. High Risk being a communicating state, this will be communicated to TTDM. Subsequently, TTDM has taken necessary steps to adjust its own state accordingly (fig. 10).

## *6. Final Discussion*

In the scope of this research, we have further demonstrated the capability of modelling diseases with SXMDM and introduced CSXM to overcome the previously identified limitation of not being able to represent the relationship between diseases [3]. Even though we have overcome one of the limitations that we identified through our previous research, still this is lacking the user interface to be recognised as a tool for generic users and still lacking the capability of representing the inner states (substages of diseases). However, SXMCM, with combining the SXMDM (TTDM & CVDRM) has demonstrated the capability of modelling the diseases and its inter relationship which





would enable medical professionals and the researchers to model diseases with high level of accuracy.

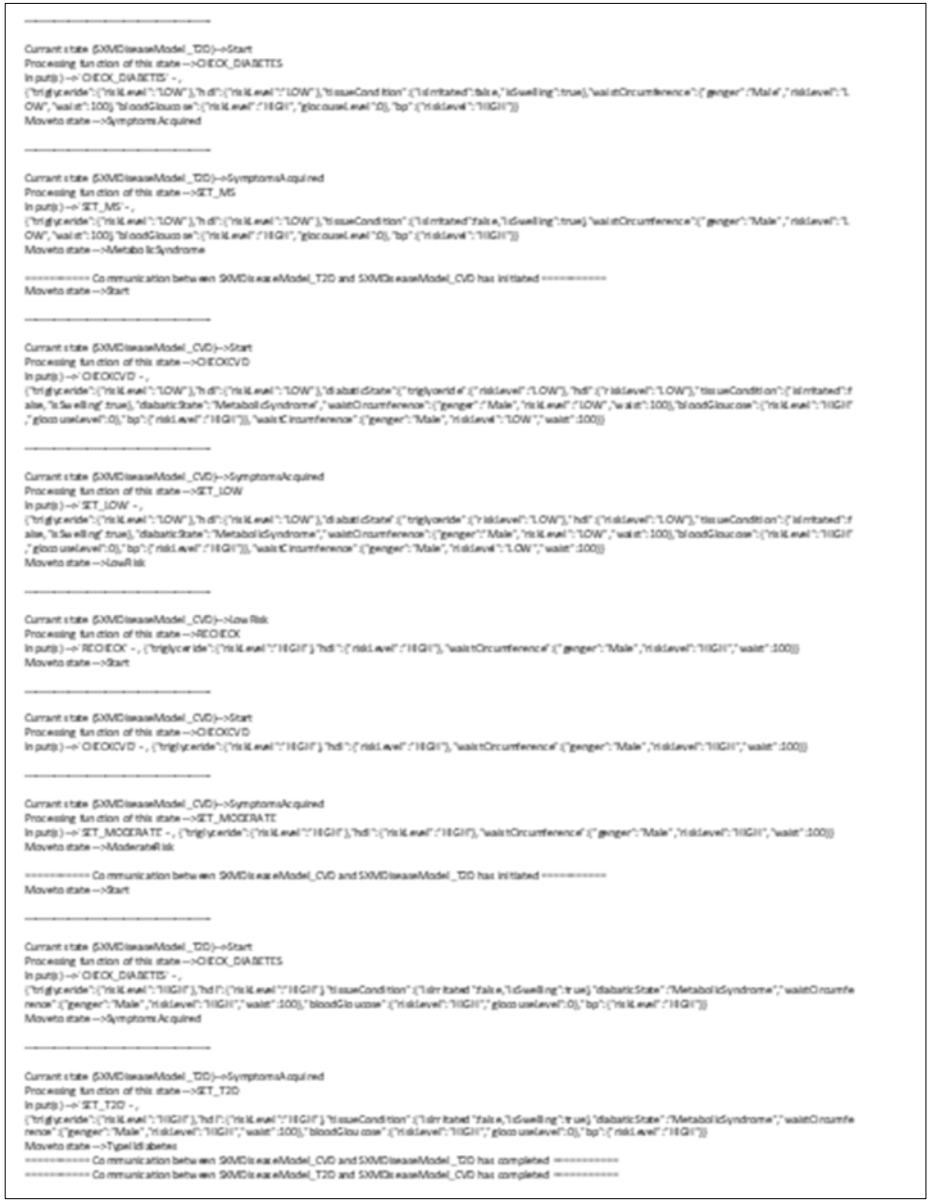


*Fig 10. Communication between CVDRM & TTDM*

In this research, using formal methods in general and especially classes of X-Machines we have identified some potential limitations. One of the most prominent would the limitation of having inner

states as mentioned above. Hence, diseases might have internal states or many stages, managing and designing could be complicated.

This has been intended to be used by professionals from various domains. The limitation from the previous model remains that, this is heavily programming dependent (all the models need to be programmed using JAVA). This poses a serious limitation in both time and expertise.

## *7. Conclusion & Future Research*

With the limitations identified in the Section 6, we the research team is intended to investigate the possibility of introducing the inner states in the SXMDM and remain the capability using SXMCM. With the successful implementation of the SXMCM, we have been able to demonstrate the ability of state base communication while preserving the identity of the particular state that communication has been initiated. Furthermore, this has given the ability of customising the response to the communication from the point of receiving the message. As discussed in the Section 2, the world is moving towards alternative medicine and lifestyle interventions for treating, managing, and preventing chronic diseases. The ultimate intention of this research is to produce a comprehensive tool which has the capability of modelling diseases and their inter-relationships and has the capability of mimicking the reaction upon exposure to nutrients. This will give the medical professionals the ability of determining the course of action without risking the lives of patients and apply alternative medicine in patient care with more confidence.

In conclusion, this paper has presented a novel approach of modelling diseases with their inter-relationship(s) without concerning the inner states (substages) of the diseases. Even though the current combination of SXMDM and SXMCM, has presented a novel approach of disease modelling with the discussed reservations in Section 6. As the main researcher in this project, we firmly believe that this is a significant milestone of our research. We are committed to conduct further research on this and improve the disease models further.

## References

[1] S. Licher, A. Heshmatollah et al. Lifetime risk and multimorbidity of non-communicable diseases and disease-free life expectancy in the general population: A population-based cohort study. PLOS Medicine, vol. 16, issue 2, 2019, article id e1002741.

[2] J.F. Ferguson, H. Allayee et al. Nutrigenomics, the Microbiome, and Gene-Environment Interactions: New Directions in Cardiovascular Disease Research, Prevention, and Treatment: A Scientific Statement from the American Heart Association. Circulation: Genomic and Precision Medicine, vol. 9, issue 3, 2016, pp. 291-313.

[3] S. Jayatilake, E. Ogunshile et al. Modelling Diseases with Stream X-Machine. In Proc. of the 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021, pp. 61-68.

[4] B.M. Sörensen et al. Prediabetes and Type 2 Diabetes are Associated with Generalized Microvascular Dysfunction: The Maastricht Study, Circulation, vol. 134, issue 18, 2016, pp. 1339–1352.

[5] S.F. Hossain, M. Huang et al. Inter Disease Relations Based on Human Biomarkers by Network Analysis, In Proc. of the IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), 2019, pp. 103-108.

[6] J. Seo, B. Choi et al. The relationship between multiple chronic diseases and depressive symptoms among middle-aged and elderly populations: results of a 2009 korean community health survey of 156,747 participants. BMC Public Health, vol. 17, no. 1, 2017, article id 8442.

[7] C.D. Filippo, M. Di Paola et al. Diet, Environments, and Gut Microbiota. A Preliminary Investigation in Children Living in Rural and Urban Burkina Faso and Italy. Frontiers in Microbiology, vol. 8, 2017, article id. 1979

[8] E.A. Finkelstein, J.G. Trogdon et al. Annual Medical Spending Attributable to Obesity: Payer-And Service-Specific Estimates. Health Affairs, vol. 28, 2009, pp. w822-–w831.

[9] J. Bowen and V. Stavridou. Safety-critical systems, formal methods and standards. Software Engineering Journal, vol. 8, issue 4, 1993, pp. 182-183.





[10] J.S. Ostroff. Formal methods for the specification and design of real-time safety critical systems. Journal of Systems and Software, vol. 18, issue 1, 1992, pp. 33-60.

[11] S. Gerhart, D. Craigen, and T. Ralston. Experience with formal methods in critical systems. IEEE Software, vol. 11, issue 1, 1994, pp. 21-28.

[12] S. Eilenberg, Automata, languages, and machines. Academic Press, 1974, 450 p.

[13] F. Ipate and M. Holcombe. A method for refining and testing generalised machine specifications. International Journal of Computer Mathematics, vol. 68, issue 3-4, 1998, pp. 197-219.

[14] F. Ipate, M. Holcombe. Specification and Testing Using Generalised Machines: A Presentation and a Case Study. Software Testing, Verification and Reliability, vol. 8, issue 2, 1998, pp. 61-81.

[15] M.A. Boemo, L. Cardelli, and C.A. Nieduszynski. The Beacon Calculus: A formal method for the flexible and concise modelling of biological systems. PLOS Computational Biology, vol. 16, issue 3, 2020, article id. e1007651.

[16] I. Stamatopoulou, I. Sakellariou et al. Formal modelling for in-silico experiments with social insect colonies. In Proc. of the 11th Panhellenic Conference in Informatics (PCI'07), vol. B, 2007, pp. 79-89.

[17] S. Coakley, R. Smallwood, and M. Holcombe. Using x-machines as a formal basis for describing agents in agent-based modelling. In Proc. of the Spring Simulation Multiconference 2006 (SpringSim'06), 2006, pp. 33-40.

[18] T. Balanescu, A. Cowling et al. Communicating Stream X-Machines Systems are no more than X-Machines. Journal of Universal Computer Science, vol. 5, issue 9, 1999, pp. 494-507.

[19] H. Georgescu and C. Vertan. A New Approach to Communicating X-Machine Systems. Journal of Universal Computer Science, vol. 6, issue 5, 2000, pp. 490-502.

[20] J. Barnard. COMX: a design methodology using communicating X-machines. Information and Software Technology, vol. 40, issue 5-6, 1998, pp. 271-280.

[21] P. Kefalas, G. Eleftherakis, and E. Kehris. Communicating X-Machines: From Theory to Practice. Lecture Notes in Computer Science, vol. 2563, 2003, pp. 316-335.

[22] M.A. Schaub, T.A. Henzinger, and J. Fisher. Qualitative networks: a symbolic approach to analyze biological signaling networks. BMC Systems Biology, vol. 1, 2007, article no. 4.

[23] A. Naldi, D. Thieffry, and C. Chaouiya. Decision Diagrams for the Representation and Analysis of Logical Models of Genetic Networks. Lecture Notes in Computer Science, vol. 4695, 2007, pp. 233-247.

[24] N. Miskov-Zivanov, P. Wei, and C. S. C. Loh. THiMED: Time in Hierarchical Model Extraction and Design. Lecture Notes in Computer Science, vol. 8859, 2014, pp. 260-263.

[25] S. Wolfram. A New Kind of Science. Available at: https://www.wolframscience.com/, accessed May 25, 2021.

[26] I. Stamatopoulou, M. Gheorghe, and P. Kefalas. Modelling Dynamic Organization of Biology-Inspired Multi-Agent Systems with Communicating X-Machines and Population P Systems. Lecture Notes in Computer Science, vol. 3365, 2005, pp. 389-403.

[27] A. Rashid, O. Hasan et al. Formal reasoning about systems biology using theorem proving. PLOS ONE, vol. 12, issue 7, 2017, article id. e0180179.

[28] Q. Wang. Formal Methods for Biological Systems: Languages, Algorithms, and Applications. Ph.D. Thesis. Carnegie Mellon University, 2016, 149 p.

[29] Q. Wang and E. M. Clarke. Formal modeling of biological systems. In Proc. of the 2016 IEEE International High Level Design Validation and Test Workshop (HLDVT), 2016, pp. 178-184.

[30] J. Mortimer, ed. The FMS Report: Ingersoll Engineers. Springer, 1984, 180 p.

[31] C. Chaouiyaю Petri net modelling of biological networksю Briefings in Bioinformatics, vol. 8, issue. 4, 2007, pp. 210-219,

[32] T. Koster, P. J. Giabbanelli, and A. Uhrmacher. Performance and Soundness of Simulation: A Case Study Based on a Cellular Automaton for In-Body Spread of HIV. In Proc. of the 2020 Winter Simulation Conference (WSC), 2020, pp. 2281-2292.

[33] G. Pardini. Formal modelling and simulation of biological systems with spatiality. Ph.D. Thesis. Università di Pisa, 2011, 145 p.

[34] D. Dranidis, K. Bratanis, and F. Ipate. JSXM: A Tool for Automated Test Generation. Lecture Notes in Computer Science, vol. 7504, 2012, pp. 352-366.

[35] W. Hao, H.M. Komar et al. Mathematical model of chronic pancreatitis. Proceedings of the National Academy of Sciences *(*PNAS), vol. 114, issue 19, 2017, pp. 5011-5016.

[36] M.B. Alazzam, A.A. Hamad, and A.S. AlGhamdi. Dynamic Mathematical Models' System and Synchronization. Mathematical Problems in Engineering, 2021, article id. 6842071.

[37] A. De Gaetano, T. Hardy et al. Mathematical models of diabetes progression. American Journal of Physiology-Endocrinology and Metabolism, vol. 295, issue 6, pp. E1462–E1479.
[38] D. Wodarz and M.A. Nowak. Mathematical models of HIV pathogenesis and treatment. BioEssays, vol. 24, issue 12, 2002, pp. 1178-1187.
[39] M. Buysschaert, J.-L. Medina et al. Definitions (and Current Controversies) of Diabetes and Prediabetes. Current Diabetes Reviews, vol. 12, issue 1, 2015, pp. 8-13.
[40] S. M. Haffner. Pre-diabetes, insulin resistance, inflammation and CVD risk. Diabetes Research and Clinical Practice, vol. 61, 2003, pp. S9-S18.

## Information about authors / Информация об авторах

Senerath Mudalige Don Dilshan Dhanishtha JAYATILAKE, Master of Science. Research interests: Software Development, Java Programming, Object-Oriented Programming, Java Language, Relational Databases.

Сенерат Мудалиге Дон Дилшан Дхаништха ДЖАЯТИЛАКЕ, магистр наук. Область научных интересов: разработка программного обеспечения, программирование на Java, объектно-ориентированное программирование, язык Java, реляционные базы данных.

Khoa PHUNG, PhD Student. Research interests: Java, program testing, software fault tolerance, decision trees, diseases, learning (artificial intelligence), medical computing, medical diagnostic computing, mobile computing, multilayer perceptrons, pattern classification, program debugging, quality assurance, software metrics, software quality, support vector machines.

Хоа ФУНГ, аспирант. Область научных интересов: Java, тестирование программ, отказоустойчивость программного обеспечения, деревья решений, болезни, обучение (искусственный интеллект), медицинские вычисления, медицинские диагностические вычисления, мобильные вычисления, многослойные персептроны, классификация шаблонов, отладка программ, обеспечение качества, метрики программного обеспечения, качество программного обеспечения, машины опорных векторов.

Emmanuel OGUNSHILE, Ph.D., Senior Lecturer in Computer Science and Chair Athena SWAN process. Research interests: Automated Specification, Verification and Testing of Software and Hardware, Model Based System Engineering, Formal Methods, Object-Oriented Languages, Type Theory, Cloud Computing, Big Data, Machine Learning, Robotics, Artificial Intelligence, Data Science, Internet of Things, Cyber-Security and Natural Language Processing.

Эммануэль ОГУНШИЛЕ, кандидат наук, старший преподаватель компьютерных наук. Научные интересы: автоматизированная спецификация, проверка и тестирование программного и аппаратного обеспечения, разработка систем на основе моделей, формальные методы, объектно-ориентированные языки, теория типов, облачные вычисления, большие данные, машинное обучение, робототехника, искусственный интеллект, наука о данных, Интернет вещей, кибербезопасность и обработка естественного языка

Mehmet AYDIN, Ph.D., Senior Lecturer in Computer Science. Research interests: machine learning, particularly reinforcement learning, wired/wireless network planning and optimization, combinatorial optimization, parallel and distributed metaheuristics, evolutionary computation and intelligent agents and multi agent systems.

Мехмет АЙДИН, кандидат наук, старший преподаватель компьютерных наук. Область научных интересов: машинное обучение, особенно обучение с подкреплением, планирование и оптимизация проводных/беспроводных сетей, комбинаторная оптимизация, параллельная и распределенная метаэвристика, эволюционные вычисления, интеллектуальные агенты и мультиагентные системы.