Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

The 9th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2019)
November 4-7, 2019, Coimbra, Portugal

SmartCGMS as an Environment for an Insulin-Pump Development with FDA-Accepted In-Silico Pre-Clinical Trials

Martin Ubl[a]*, Tomas Koutny[b]

[a]*Department of Computer Science and Engineering, University of West Bohemia, Technicka 8, 306 14 Plzen, Czech Republic*
[b]*NTIS – New Technologies for Information Society, University of West Bohemia, Technicka 8, 306 14 Plzen, Czech Republic*

Abstract

Diabetes is a widespread civilization disease. It manifests with an elevated blood glucose level. In the long-term, elevated blood glucose level continuously damages organs. In the short-term, hypo- and hyperglycemia are acute complications. Insulin lowers blood glucose level by promoting its utilization. At basal rate, insulin pump delivers insulin to subcutaneous tissue to control blood glucose level. In addition, patient doses insulin boluses in accordance with estimated carbohydrate content of consumed meal. Control algorithm of the pump considers the boluses, when calculating the basal rate. In our previous work, we have proposed a parallel-architecture for the next-generation of glucose monitoring - SmartCGMS. It unifies the source-code base and the glucose-monitoring-and-control paradigm across real, simulated and prototyped devices. As the development continues, especially towards the pump-control algorithms, we face a problem of reducing the SmartCGMS requirements when considering a low-power hardware. In this paper, we present the modifications that lead to a reduced number of threads, while implementing the closed-loop feedback between a glucose sensor and insulin pump to conduct FDA accepted *in-silico* pre-clinical trials.

Keywords: diabetes, glucose, monitoring, insulin pump, in-silico, pre-clinical trial

* Corresponding author. Tel.: +420 377 632 453; fax: +420 377 634 402.
E-mail address: ublm@kiv.zcu.cz

1. Introduction

Diabetes mellitus is a widespread civilization disease. It is a heterogonous group of diseases, which share the same symptom – elevated blood glucose level (BG) [1]. There is either a relative (diabetes type 2) or absolute insulin insufficiency – diabetes type 1 (T1D). While using of insulin pump is pre-dominant with T1D, recent research suggests the pump to treat both types of diabetes [2].

Cells use glucose as a fuel to produce energy. Insulin moderates glucose utilization. Therefore, insulin pump delivers insulin to promote glucose utilization. Thus, it reduces BG as elevated BG causes both acute and chronical complications.

Glucose-level sensor comprises a needle installed in subcutaneous tissue. The needle measures electric current, produced by a glucose-triggered chemical reaction. Then, it uses reference BG to interpret the measured current as a glucose level (IG) [3]. Patient obtains the reference BG with a finger prick. Due to the associated discomfort, patient measures BG two or three times a day [4]. Therefore, BG is sporadic and IG is frequent. Continuous glucose monitoring systems (CGMS) usually reports IG every five minutes.

CGMS complements insulin pump. The pump calculates insulin basal-rate. Artificially, it mimics a healthy-patient's pancreas function as the diabetic-patient's pancreas may deliver too small or none amount of insulin. This is a continuous subcutaneous insulin infusion (CSII). When ingesting the meal, the patient estimates its carbohydrate content (CHO) and delivers an additional bolus of insulin to compensate time-distributed CHO appearance in the blood [5].

When developing a new insulin-pump control algorithm, it should be tested *in-silico* prior a clinical trial. There is an FDA-accepted T1D patient simulator [6]. We have already developed a glucose control and monitoring architecture – SmartCGMS. It unifies the source-code base across real, simulated and prototyped devices [7]. In this paper, we present modifications to this architecture that allow us to conduct the FDA-accepted *in-silico* pre-clinical trials. In addition, the modifications reduce the hardware requirements to facilitate an easier transition of newly developed algorithms to a real insulin pump, i.e.; a low-power device with hard-real time scheduling [8].

1.1. SmartCGMS

SmartCGMS builds on top of the High-Level Architecture (HLA) principles [9]. HLA is a well-established simulation paradigm on which we can run a co-simulation. Co-simulation comprises a number of real, simulated and prototyped devices working together.

The SmartCGMS architecture comprises a number of linearly connected filters. Each filter has a specific functionality. For example, input filter reads BG and IG from a sensor or a database of previously measured CGMS profiles. Then, a processing filter calculates an insulin bolus from a glucose-level signal. Asynchronous filter runs in a dedicated thread. A pipe connects two adjacent filters. Such a pipe uses concurrent queue internally, thus establishing thread synchronization. Filter processes a sequence of events, described with an abstract data type [7]. Each event contains GUID identifier [10] to designate the particular physiological signal it represents. Filter receives the event from a pipe on its input, processes the event and writes it to the pipe on its output. In addition, a filter can generate additional events as needed.

Another type of filter can control a medical device such as insulin pump, thus becoming a CSII-controller. Fig. 1 depicts a possible insulin pump setup. In this setup, we read IG from CGMS sensor. Using exponential [11] and absorption [12] models, we calculate insulin on board (IOB) and CHO on board (COB) respectively. Both models produce events, whose GUIDs associate these signals with these particular models. By rewriting GUID identifiers of these events, we remap them to represent the respective physiological signal. We do so because the subsequent filters expect these physiological signals, which we have to estimate because we cannot measure them in the practice.

If patient announces a meal with a user-interface filter, a specific filter calculates insulin bolus. It calculates the bolus from the estimated CHO content of the meal. The user-interface filter creates an event that describes the CHO content. This event passes through all the subsequent filters. Therefore, the COB filter receives it and updates COB accordingly.

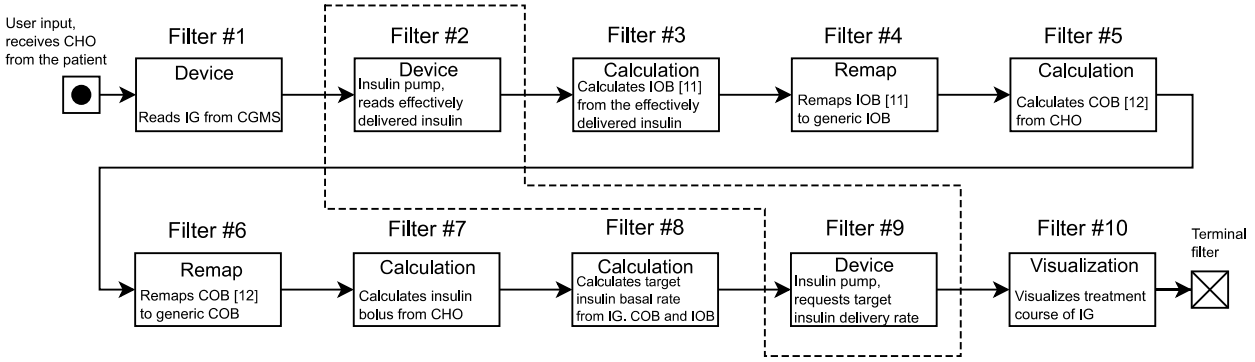

Figure 1: SmartCGMS configured to control insulin pump with the original architecture.

Another filter calculates the insulin basal-rate from IG, COB and IOB. This basal-rate signal propagates to an insulin-pump device-control filter. Eventually, we visualize the entire glucose-level management.

The entire chain of filters terminates with a specific filter that consumes all the events, but forwards none.

As the original SmartCGMS architecture considered all filters as asynchronous [7], the number of threads increased with each added filter. By considering future transition to embedded devices, we needed to reduce the computational requirements of the original architecture. Section 2 proposes the respective modifications.

1.2. Related Work

Pre-clinical trial precedes the clinical trial to collect and analyze feasibility and safety data. With insulin-pump development, there is the possibility of *in-silico* testing. Such a testing uses a patient model, e.g.; Bergman model [13], Hovorka model [14] or an FDA-accepted one. A number of projects partially adapted the Bergman and Hovorka models to build a simulation environment. Such an environment is e.g. Physiomodel [15], SimEdu [16], and T1DMS [6]. Nevertheless, they do not provide a simulation of the insulin-pump software stack.

Physiomodel [15] and SimEdu [16] provide a number of models for different human body systems. Nevertheless, no official regulatory body recognizes them. In addition, they are not suitable for a co-simulation with a different environment.

T1DMS is an FDA-accepted T1D patient simulator. It provides a cohort of T1D patients and a number of FDA-accepted scenarios. It allows a coupling with a custom insulin pump-controller. Simulink block comprises such a controller. Then, we replace such a block with a different component, e.g.; with SmartCGMS, while adhering to the Simulink-block interface specification.

A number of various studies conducted pre-clinical *in-silico* trials using T1DMS [17, 18, 19]. None of them considered a co-simulation with an insulin-pump environment. Therefore, we develop the SmartCGMS to fill this gap.

2. The Proposed SmartCGMS Modifications

2.1. Reducing the Parallelism

By dedicating a thread to each filter, we obtained asynchronous filter. While such a filter has a low-complexity interface to the system, it introduces an additional overhead if the filter performs a synchronous operation. For example:

- The remapping filter checks event's GUID and rewrites it, if it matches the given settings
- Using a given model of glucose-dynamics, the calculation filter produces new events based on the input events
- A log filter stores the events to an external storage

With filters like these, input event triggers the filter to generate its output. It is a synchronous behavior. By having all filters executed asynchronously, processing an event requires a transition to the kernel space of the underlying operating system. Embedded device would benefit from a reduced number of threads [20]. On the other hand, source-code maintenance profits from the low-complexity of the original SmartCGMS architecture. Therefore, we introduce two different pipe implementations, synchronous and asynchronous, with the same interface. Subsequently, we implement synchronous filters in addition to the original, asynchronous filters. Synchronous filter executes in the context of the nearest preceding asynchronous filter. The synchronous pipe implies this.

From the output of asynchronous filter, the synchronous pipe creates a vector of events. Initially, the vector contains a single event, produced by the asynchronous filter. Then, the synchronous pipe passes this vector to the subsequent synchronous filters. Each synchronous filter can modify the vector. Hence, this entire activity executes within the pipe's Send function, which the preceding asynchronous filter executes. Eventually, a succeeding asynchronous filter receives the events from the vector sequentially. Fig. 2 depicts such an event processing.

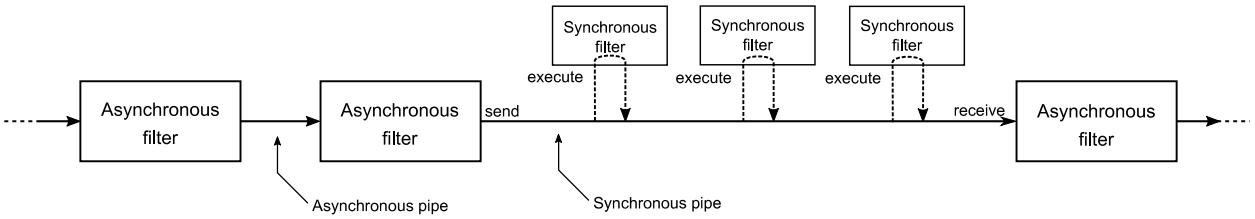

Figure 2: Executing synchronous filters in a thread-context of the nearest preceding asynchronous filter

2.2. In-Silico Closed Loop

A pre-clinical study should precede a clinical trial. For this purpose, there is an FDA-accepted simulator – T1DMS. It enables evaluation of CSII controllers [6]. The simulator calls a custom-defined function. Function arguments are BG, IG, meals announcement and currently delivered volume of insulin. The simulator expects this function to return insulin bolus and basal-rate values.

Following the High-Level Architecture principles, it is important to maintain a single code base to reduce differences between simulation, laboratory and real-world deployment. As depicted in Fig. 1, there are two filters, which connect to the same insulin pump. General behavior of an insulin pump implies this. When requesting a particular insulin delivery rate, the pump may reject or modify the rate according to its internal logic and current conditions. Therefore, we must read the effectively set insulin delivery-rate and have it available before calculating IOB. As a result, there must be two filters connected with a feedback link.

The patient enters the estimated CHO content of a consumed meal. From CHO, we calculate COB and insulin bolus. From the effectively delivered insulin volume, advertised with the feedback link, we calculate IOB. From IG, IOB and COB, we calculate target insulin basal delivery rate. On announced meal, we request the pump to deliver the calculated insulin bolus. This affects the effectively delivered volume of insulin, which we advertise in the feedback link.

IG changes because of delivering the insulin to the (real or simulated) subcutaneous tissue. This closes the loop. Specifically, by enabling the patient's CHO input, we support a hybrid closed loop system. In the original architecture, there was no feedback link as we regulated IG based on IG only.

With the feedback link, we need to preserve time ordering of the events. A filter that receives events from the feedback link maintains a priority queue. Once this filter emits an event, it buffers all incoming events in the priority queue. Following the emitted event, it sends additional, synchronization event. Once the synchronization event reaches a filter, which issues commands to the insulin pump, the feedback link advertises the synchronization and discards the event. Then, the respective filter removes the oldest event from the priority queue and sends it, while issuing another synchronization event after it. Thus, the synchronization events act as a heartbeat.

Time compression of the *in-silico* simulation requires the heartbeat. It synchronizes the insulin-pump feedback to the IG measurement.

3. Experimental setup

We demonstrate the proposed modifications, using two experimental setups. The first setup corresponds with the original study [7]. Fig. 4 depicts this. First filter replays a previously measured CGMS profile. Filter #2 calculates continuous BG from IG and sporadically measured BG, using the Steil-Rebrin model. Filters #3 and #4 remap the newly calculated continuous BG model signal to a measured BG, while mapping the original BG to an auxiliary, virtual signal. Then, filter #5 predicts IG based on continuous BG and IG. Filters #6 and #7 generate data for the user interface. Fig. 5 depicts this setup with the synchronous filters.

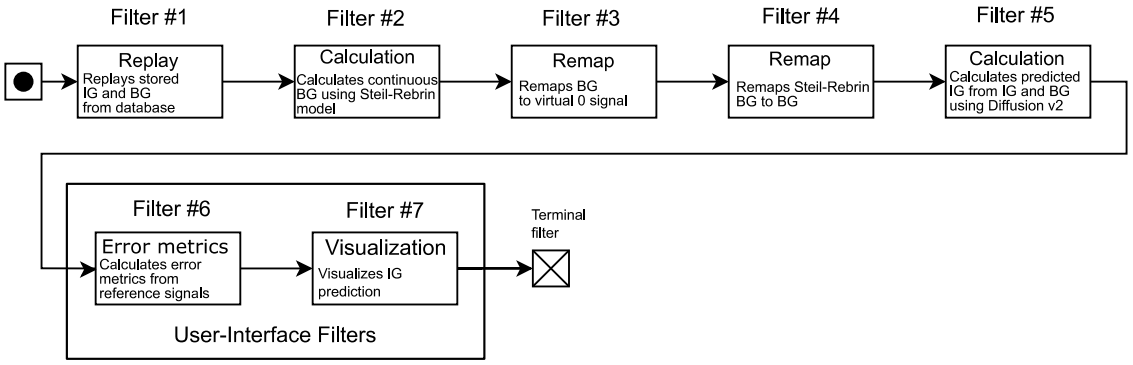

Figure 3: Experimental setup for scenario #1 – IG prediction, the original architecture

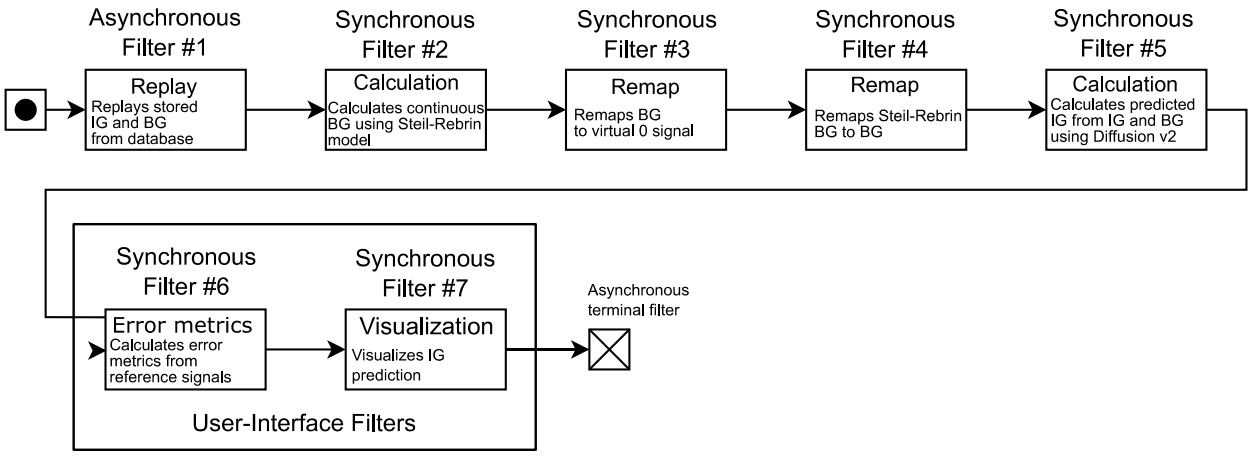

Figure 4: Experimental setup for scenario #1 – IG prediction, the modified architecture

Section 1.1 describes the second experimental setup. Fig. 1 depicts this setup with the original architecture. Fig. 5 depicts it with the proposed modifications. For the sake of simplicity, we used T1DMS with FDA-accepted scenario for a closed-loop system evaluation, with a single 50g CHO meal and *in-silico* subject, identified with the *adult#average* tag. T1DMS announced the meal. Intentionally, we configured the bolus calculator to deliver an excessive bolus of insulin. A successfully working co-simulation would exhibit a glucose-level decline below the steady glucose level. Hence, it serves the purpose of verification against the current knowledge of glucose homeostasis.

4. Results

Table 1 gives the number of threads required before and after an implementation of the proposed modifications. The proposed modification considerably decreases the number of threads, thus leading to reduced computational complexity as considerably less synchronization affects the entire processing. Let us note that computational complexity does not concern input size only – i.e.; the measured levels. In our case, it concerns the synchronization overhead as well.

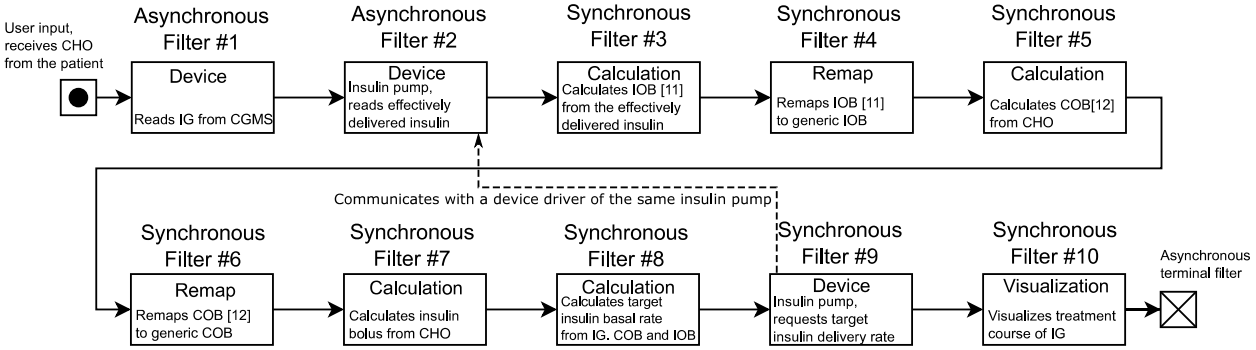

Figure 5: Experimental setup for scenario #2 – insulin pump control, the modified architecture.

Table 1. Thread count in the originally proposed state and with modified code

	Original architecture	Proposed modifications
Scenario #1	8	2
Scenario #2	11	3

Regarding the second experimental setup, SmartCGMS determined the basal insulin rate as 1.2 U per hour. It has not changed during the experiment. CHO produced a peak above the steady IG. Then, IG declined below the steady IG after delivering the excessive insulin bolus. After that, IG recovered the steady level. Fig. 6 depicts this.

As IG declined below the steady state, it demonstrates that the feedback link performs correctly and that we have closed the loop successfully.

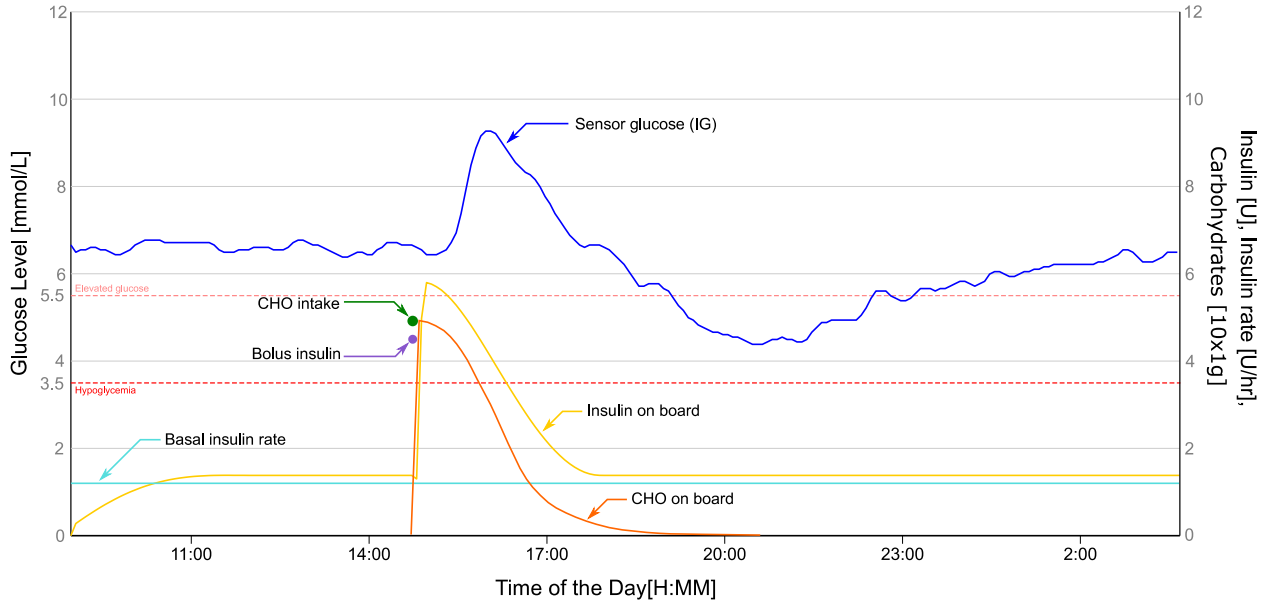

Figure 6: Simulation with T1DMS demonstrating the closed loop

5. Conclusion

We proposed and implemented two modifications to the original SmartCGMS architecture. The first one reduced the computational requirements. The second modification established a feedback link needed for an adaptive calculation of insulin delivery volume, when optimizing diabetes type 1 treatment with CSII. We successfully verified both modifications with FDA-accepted simulator and FDA-accepted experimental scenario.

As with the original architecture, we will release the modified SmartCGMS at diabetes.zcu.cz [21].

The future work will concern implementation details for embedded devices with a real-time scheduling.

Acknowledgements

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports and university specific research project SGS-2019-016.

References

[1] Kasper DL, Fauci AS, Hauser SL, Longo DL, Jameson JL, Loscalzo J. (2015). Harrison's principles of internal medicine (19th edition.). New York: McGraw Hill Education. ISBN: 007174889X

[2] Bode BW. Insulin pump use in type 2 diabetes. Diabetes Technologies and Therapeutics. 2010. 12 (1), pp. 17–21. doi:10.1089/dia.2009.0192

[3] Facchinetti A. Continuous Glucose Monitoring Sensors: Past, Present and Future Algorithmic Challenges. Sensors (Basel). 2016. 16(12): 2093. Published 2016 Dec 9. doi:10.3390/s16122093

[4] Murata T, Tsuzaki K, Yoshioka F, et al. The relationship between the frequency of self-monitoring of blood glucose and glycemic control in patients with type 1 diabetes mellitus on continuous subcutaneous insulin infusion or on multiple daily injections. J Diabetes Investig. 2015;6(6):687–691. doi:10.1111/jdi.12362

[5] Slattery D, Amiel SA, Choudhary P. Optimal prandial timing of bolus insulin in diabetes management: a review. Diabet Med. 2017. 35(3):306–316. doi:10.1111/dme.13525

[6] Man CD, Micheletto F, Lv D, Breton M, Kovatchev B, Cobelli C. The UVA/PADOVA Type 1 Diabetes Simulator: New Features. J Diabetes Sci Technol. 2014;8(1):26–34. doi:10.1177/1932296813514502

[7] Koutny T, Ubl M. Parallel software architecture for the next generation of glucose monitoring. Procedia Computer Science, 141, 279-286, 2018. doi: 10.1016/j.procs.2018.10.197

[8] Zhang Y, Jetley R, Jones PL, Ray A. Generic safety requirements for developing safe insulin pump software. J Diabetes Sci Technol. 2011. 5(6):1403–1419. Published 2011 Nov 1. doi:10.1177/193229681100500612

[9] Topcu O, Durak U, Qguztuzun Hm Yilmaz L. High level architecture. Distributed Simulation. Simulation Foundations, Methods and Applications, Cham:Springer 2106.

[10] Leach P, Mealling M, Salz R. A universally unique identifier (UUID) URN namespace. RFC 4122.

[11] Turnheim K, Waldhäusl WK. Essentials of insulin pharmacokinetics. Wien Klin Wochenschr. 1988 Feb;100(3) 65-72. PMID: 3281377.

[12] Dalla CM, Camilleri M, Cobelli C. A System Model of Oral Glucose Absorption: Validation on Gold Standard Data. IEEE Transactions on Biomedical Engineering, 53(12), pp. 2472-2478, Dec. 2006. doi: 10.1109/TBME.2006.883792

[13] Bergman RN, Ider YZ, Bowden CR, Cobelli C. Quantitative estimation of insulin sensitivity. The American Journal of Physiology, 236(6), pp. 667-677, 1979. doi: 10.1152/ajpendo.1979.236.6.E667

[14] Hovorka R, Canonico V, Chassin LJ, Haueter U, Massi-Benedetti M, Orsini Federici M, Pieber TR, Schaller HC, Schaupp L, Vering T, Wilinska ME. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. Physiological measurement, 25(4), pp. 905-920, 2004. doi: 10.1088/0967-3334/25/4/010

[15] Matejak M, Kofranek J. Physiomodel – an integrative physiology in Modelica. 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 2015, Milan, 2015, pp. 1464-1467. doi: 10.1109/EMBC.2015.7318646

[16] Wilinska ME, Chassin LJ, Acerini CL, Allen JM, Dunger DB, Hovorka R. Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes. J Diabetes Sci Technol. 2010;4(1):132–144. Published 2010 Jan. doi:10.1177/193229681000400117

[17] DeJournett L, DeJournett J. In Silico Testing of an Artificial-Intelligence-Based Artificial Pancreas Designed for Use in the Intensive Care Unit Setting. J Diabetes Sci Technol. 2016;10(6):1360–1371. Published 2016 Jun 14. doi:10.1177/1932296816653967

[18] Campos-Náñez E, Layne JE, Zisser HC. In Silico Modeling of Minimal Effective Insulin Doses Using the UVA/PADOVA Type 1 Diabetes Simulator. J Diabetes Sci Technol. ;12(2):376–380. doi:10.1177/1932296817735341

[19] Breton MD, Hinzmann R, Campos-Nañez E, Riddle S, Schoemaker M, Schmelzeisen-Redeker G. Analysis of the Accuracy and Performance of a Continuous Glucose Monitoring Sensor Prototype: An In-Silico Study Using the UVA/PADOVA Type 1 Diabetes Simulator. J Diabetes Sci Technol. ;11(3):545–552. doi:10.1177/1932296816680633

[20] Davis R, Merriam N, Tracey N. How embedded applications using an RTOS can stay within on-chip memory limits. 12th EuroMicro Conference on Real-Time Systems, pp. 71-77. June 2000.

[21] Koutny T, Krcma M, Kohout J, Jezek P, Varnuskova J, Vcelak P, Strnadek J. On-line Blood Glucose Level Calculation. Procedia Computer Science, 98, pp. 228-235, 2016. doi: 10.1016/j.procs.2016.09.037