Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 6441–6446

Object-oriented modelling of advanced computer cooling solutions

Alberto Leva * **Federico Terraneo** * **Tobia Cancelliere** ** **Marco Chioggi** ** **William Fornaciari** * **David Atienza** ***

* *DEIB, Politecnico di Milano, Italy (e-mail: {alberto.leva,federico.terraneo,william.fornaciari}@polimi.it).*
** *Former graduate student at the DEIB.*
*** *Embedded Systems Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland (e-mail david.atienza@epfl.ch).*

Abstract: Modern computing systems are so energy-intensive to make efficient cooling vital for their operation. This is giving rise to a variety of innovative cooling solutions based on a mix of traditional and new techniques. The design and engineering of these solutions, as well as of the necessarily involved controls, requires dynamic simulation. Cooling simulation models must be capable of representing multi-physics cyber-physical systems, of connecting to specialised 3D chip simulators when high detail is needed, and at the same time of scaling up to the data centre – tailoring the detail level accordingly – when system-level studies need carrying out. In such a challenging *scenario*, an enabling technology is Object-Oriented Modelling (OOM). Along this approach we here present a Modelica library to serve the purposes just outlined, and that we are releasing as free software for the scientific and engineering community.

Keywords: Computer cooling, dynamic modelling, object-oriented modelling simulation, multi-physics systems, cyber-physical systems, model-based engineering.

1. INTRODUCTION

Modern computing equipment calls for a new generation of heat dissipation systems. The result is the development of several technologies such as liquid cooling, evaporative systems, Peltier elements, and numerous combinations thereof — that is, of articulated and often multi-physics cooling solutions that scale from the single device up to the data centre level, see for example the recent reviews by Aglawe et al. (2022) and Zhang et al. (2021).

Given the impressive power density of modern processors, nowadays, an incorrect behaviour of the cooling system can not only impact the performance and the long-term reliability of a computing device, but even jeopardise its integrity. As a consequence, first any heat dissipation solution has to comprehend some controls, and then a model-based design of such solutions (controls included) is mandatory.

Quite evidently, the *scenario* just sketched requires game-changing modelling and simulation capabilities. The library presented in this paper aims to be a contribution in this direction.

The rest of the paper is organised as follows. Section 2 briefly reviews relevant literature so as to motivate the modelling needs and the chosen approach, subsequently outlined in Section 3 together with the innovative technology mix on which it is based. Section 4 provides an overview of the presented library, while Section 5 reports a couple of representative examples to evidence its capabilities and potential. Finally, Section 6 concludes the paper with some remarks on the activity carried out so far, also illustrating future research and application-oriented work.

2. RELATED WORK AND MOTIVATION

Computing equipment cooling involves phenomena at very different time scales, from the minutes or more taken by room- or facility-level conditioning machinery to respond, down to the millisecond one of the thermal phenomena that occur inside a microprocessor (Terraneo et al., 2019). As such, any modelling and simulation tool aimed to assess modern cooling systems must be capable – when needed – to simulate the (slower) dynamics of that system together with the said very fast phenomena, and sometimes also together with the involved thermal/power/performance policies aboard the chip, see e.g. Leva et al. (2017); Ali et al. (2022). Indeed, modelling and simulation of heat dissipation systems nowadays need a qualitative leap from several viewpoints, most notably computation speed, accuracy and model maintainability.

The difficulties to face are numerous. Problems are inherently cyber-physical, owing to the presence of hydraulics and thermodynamics (Seuret et al., 2018) coupled with cooling circuit control algorithms and on-chip policies, to provide the physical and the cyber part, respectively. In addition, the physical part is always multi-domain, and exhibits a variety of configurations that is so vast to make a component-based approach a must. Also, the systems to simulate can be of large size, due to the frequent need for fine-grained spatial resolutions. At the same time, finally, the level of modelling detail must be scalable to achieve

10.1016/j.ifacol.2023.10.856

the maximum computational efficiency in each simulation study.

Traditional thermal simulation approaches very often prove unsuitable for such new operating *scenarii.* The said approaches are based on exploiting the particular characteristics of dynamic thermal modelling when applied to microprocessors, while less attention is devoted to cooling equipment. Thus, the same optimisations become a limit for the range of cooling solutions that can be modelled. Hence the tools just mentioned do achieve fast simulation, but at the deliberate expense of generality. Modern alternatives such as Equation-Based Modelling (EBM) can be applied but suffer from the symmetric problem: they are naturally keen to represent the heterogeneous physics encountered in cooling systems, but pay for this capability in terms of computational efficiency.

Recently, we attempted to join the best of the two modelling approaches sketched above, making the 3D-ICE thermal simulator (Sridhar et al., 2010; Terraneo et al., 2021) capable of performing co-simulation with object-oriented, equation-based modelling and simulation tools (Terraneo et al., 2021). In this paper, we address the other side of the overall construction, that is, a library of models to provide the announced EBM counterpart to 3D-ICE. We carry out the required model development by employing the Modelica language (Modelica Association home page, 2022; P. Fritzson, 2014) and by exploiting the Functional Mock-up Interface (FMI) standard (Functional Mock-up Interface standard home page, 2022; Blochwitz et al., 2011). However, the underlying ideas are general with respect to the adopted tools.

For completeness, in addition to 3D-ICE we also mention the interesting alternative approach behind the MTA simulator (Ladenheim et al., 2018), that is based on a purpose-specific solver for the differential-algebraic system to address. In principle MTA could be made compliant with the FMI standard, thereby becoming able to take profit from the presented cooling systems library. The same applies to the HotSpot simulator (Huang et al., 2006), as well as to other alternatives that we do not mention.

3. MODELLING NEEDS AND APPROACH

A solution for modern cooling system simulation has to fulfil several needs, briefly listed and commented on below.

First, it must allow for first-principle models, based on dynamic balances, as well as for data-based models, typically identified from recorded data. In the former case models must ba a-causal, so as to represent solution-time causality variations owing, e.g., to flow reversals. Also, abstract models just providing interfaces (e.g., a component with two liquid flanges) must be provided, to be specialised either for different behaviours (pump, valve and so on) or – within the same behaviour – for different levels of detail (e.g., accounting or not for fluid compressibility).

Then, it must be capable of representing the properties and the thermodynamic behaviour of various substances, including solids (such as rack or liquid pipe walls), single-phase subcooled liquids (water, glycols, and so on), ideal gases (e.g., dry air) but moist air as well, and (not yet covered by the presented library but to be included in future releases) phase-transitioning species like refrigerating fluids. In this last case, openness to linking external substance property calculation codes such as REFPROP (Lemmon et al., 2007) is a desirable feature.

The requirements made so far entail that models of fluid network elements (the majority) are naturally partitioned into “mass storage”, “mass flow”, “energy storage” and “energy flow” ones. The mass storage type, referring to a control volume V, takes the form

$$\left\{\begin{aligned} V\frac{d\rho(p,h)}{dt} &= \sum_{i=1}^{n_p} w_i \\ V\frac{d(\rho(p,h)e(p,h))}{dt} &= \sum_{i=1}^{n_p} w_i\tilde{h}_i + \sum_{j=1}^{n_s} Q_j \\ \tilde{h}_i &= \begin{cases} h_i & w_i \geq 0 \\ h & w_i < 0 \end{cases} \end{aligned}\right. \tag{1}$$

where the fluid state is assumed to be represented by pressure p and specific enthalpy h, $\rho(p,h)$ and $e(p,h)$ are density and specific internal energy as dictated by the fluid characteristics, w_i is the mass flowrate (positive if entering v at the i-th out of n_p connection ports, h_i the specific enthalpy presented at that port by the component(s) connected from outside, and Q_j the heat rate (positive if entering) at the j-th out of n_s exchanging surfaces.

The mass flow type connects two storage elements (a and b to name them), is algebraic unless fluid inertia needs considering (which is hardly ever necessary in the addressed context), and reads

$$w_{ab} = f(p_a, p_b, h_a, h_b\theta(p_a, h_a), \theta(p_b, h_b)) \tag{2}$$

where w_{ab} is the a to b mass flowrate, $p_{a,b}$ and $h_{a,b}$ are pressures and enthalpies at a and b, and $\theta(p_{a,b}, h_{a,b})$ are the fluid properties involved in the particular flow correlation considered, depending on pressure and enthalpy as dictated by the fluid physics.

Notice in (1) and (2) the separation between model equations and fluid properties, which is made possible by the EBM paradigm. Analogous considerations could be made about energy storage (typically in solid bodies) and flow, but are omitted here for brevity.

Coming to control elements, these must be represented both as continuous-time models and as clocked digital algorithms, and the analyst must be allowed to select either representation for each individual element. In this respect, the possibility of providing algorithmic control representation by calling external code (most typically, C) is desirable.

Finally, the produced executable code must be reasonably easy to interface with other programs, most typically high-detail 3D electro-thermal semiconductor simulation codes.

The above features are required in the addressed domain, since the models to simulate can potentially range from a single CPU cooling loop to a full rack model, or even to an entire data centre. For apparent efficiency reasons, scaling up the size of the model calls for adequately reducing the detail level in its components, or at least in those components that are not the subject of the study at hand.

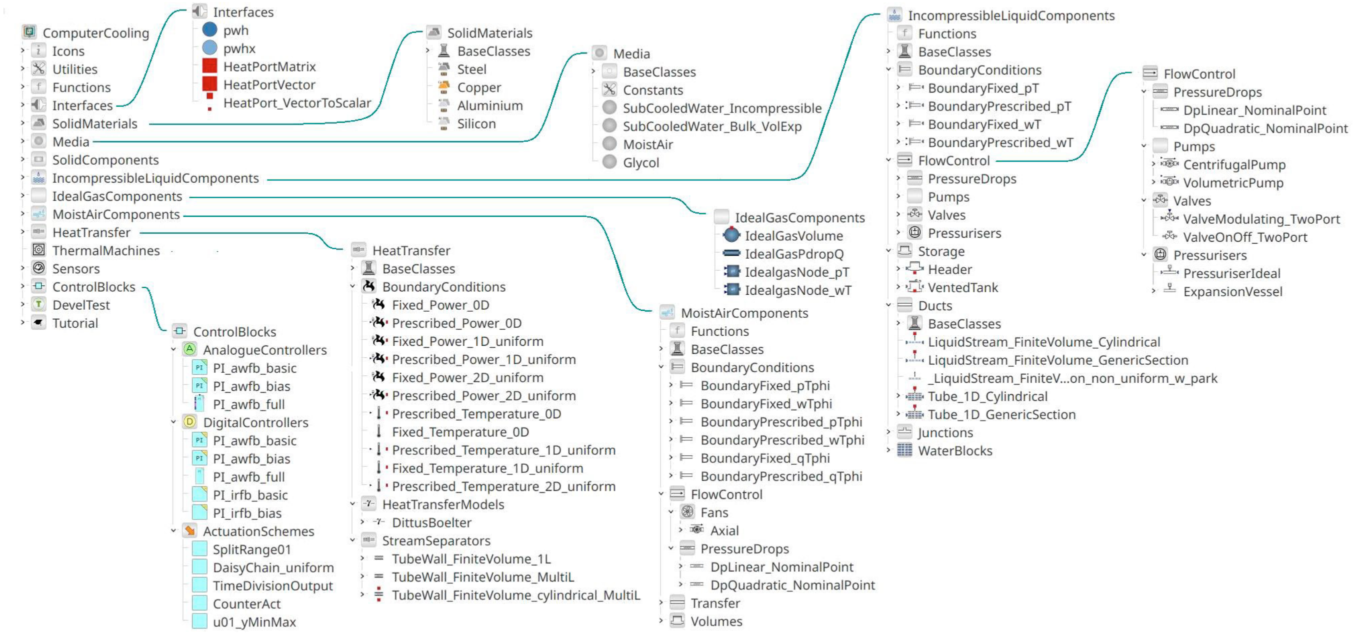

Fig. 1. Overview of the developed Modelica library.

To respond to these needs, we adopted a mix of declarative (EBM) modelling and imperative code, by developing a Modelica library capable of interfacing with both CoolProp (Bell et al., 2014) and REFPROP via the ExternalMedia one (Casella and Richter, 2008), and with chip simulators like 3D-ICE via the FMI (Functional Mockup Interface) industrial standard (Functional Mock-up Interface standard home page, 2022).

We believe this technology mix to be particularly promising in particular when coordinating on-chip thermal policies with cooling system controls, as suggested in Leva et al. (2017) and made more and more necessary by the faster chip dynamics stemming from recent lithography scales coupled to the necessity of an energy-efficient heat dissipation (and recycle when possible). In this paper we focus on the Modelica library, however; the rest of the sketched matter will be addressed in future works.

4. LIBRARY OVERVIEW

Figure 1 shows an overview of the presented library, which we develop using the free OpenModelica translator (Fritzson et al., 2020; OpenModelica Consortium home page, 2022) to maximise dissemination and experience sharing. For the same reasons, as well as to foster cooperative development, the library is available as free software within a 3-clause BSD licence at `https://github.com/looms-polimi/computer_cooling`, together with (synthetic) installation and usage instructions.

As can be seen from Figure 1, the key modelling choice is to abstract a-causal interfaces for mass exchange in the liquid or gas case (carrying pressure, mass flowrate and specific enthalpy), moist air case (with the addition of water/vapour mass fraction) and heat exchange without mass transfer (carrying temperature and heat rate) in the scalar, vector (1D, useful for piping) and matrix (2D, for contact surfaces) cases. Also, solid material and incompressible fluid properties are provided as records, while moist air is modelled based on Mollier equations, providing uniform interfaces for the above models as well to allow for easy interchangeability.

The above established, first-principle models are created along a finite-volume approach based on mass and energy equations as outlined in Section 3, including convenient correlations (such as the Dittus-Bölter one for convection) when required. Also, as anticipated, control blocks are represented both in the continuous time and as digital algorithms.

Co-simulation with 3D-ICE – or any FMI-compatible code, see the Functional Mock-up Interface standard home page (2022) for the long list of supported tools – needs just wrapping into a single component exposing input and output connectors, given the inherently causal nature of any interface with external code. Since an example of co-simulation with Modelica and 3D-ICE (not involving the library presented herein, but the procedure is analogous) was already reported in Terraneo et al. (2021), we omit here further details.

In addition, since many simulation activities concerning the design and assessment cooling systems are carried out without the need for a detailed chip simulation, care was taken to allow the library to accept recorded power data as inputs, and feed them to convenient interface components so as to present them to the cooling system model as time-varying boundary conditions. Needless to say, the export of simulation data in various formats for subsequent elaboration is possible.

Coming to validation, the activity is constantly underway, but – and this is a key point – can be done on a per-component basis, trusting then a validated component also to simulate a cooling system that does not yet exist. Ideally component validation should be done based on data collected in well-controlled laboratory condition, but this is seldom possible; an exception, relative to heat sink models is shown and discussed in Terraneo et al.

(2021), and thanks to the equation-based nature of the modelling approach employed, the obtained results in terms of exchange correlation validity were exploited when developing several other library models. In the absence of reliable data, validation can be done against analytical solutions where possible (e.g., in some stationary heat transfer cases) of versus already validated (and in general more complex and computationally heavy) models, like for example – sticking to the Modelica ecosystem – some of those in the extensively tested and thus well assessed ThermoPower library (Casella and Leva, 2006).

5. APPLICATION EXAMPLES

We now show a couple of application examples built with the developed library. The aim is to give an idea of the flexibility and ease of use of the included models in the variety of situations that cooling simulation nowadays presents, also with an eye on its use for cooling systems that involve controls.

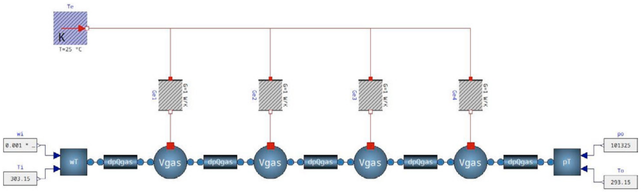

Fig. 2. Gas pipe with flow reversal – Modelica diagram.

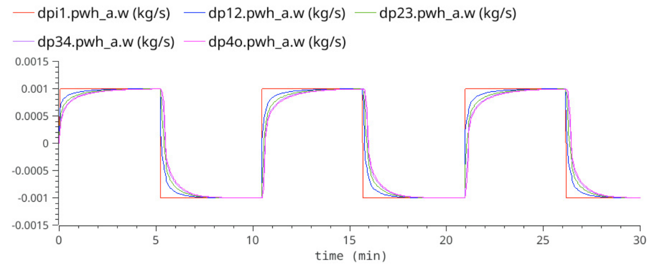

Fig. 3. Gas pipe with flow reversal – mass flowrates.

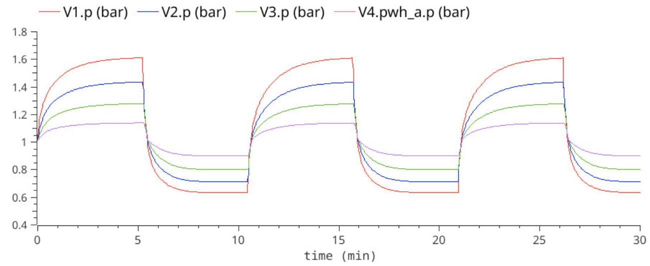

Fig. 4. Gas pipe with flow reversal – pressures.

5.1 Example 1

The first example refers to the model of an exchanging pipe containing an ideal gas, as could be the case with a cooling air duct if the effect of humidity is not to be represented (for example, to trade accuracy for computational performance or just because it is not physically relevant as it could be in a de-humidified closed circuit within a rack). The model under question is shown as Modelica diagram in Figure 2. It is composed of four volumes (storage elements) and five flow elements, while the prescribed boundary conditions are a constant pressure on the right, a varying mass flowrate on the left, and a bulk temperature for convective exchange on the outer pipe surface (the internal exchange model is embedded in the storage components).

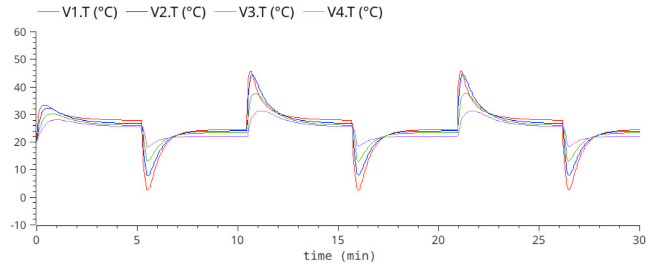

Fig. 5. Gas pipe with flow reversal – temperatures.

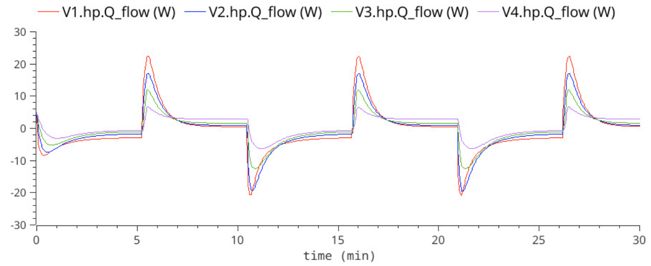

Fig. 6. Gas pipe with flow reversal – thermal powers.

Figures 3 through 6 report the results of a simulation test in which the flowrate on the left was varied so as to change sign, thereby causing causality reversals in the model. The effect on inter-volume flowrates and pressures evidences that compressibility is properly accounted for (see figures 3 and 4).

Since the two boundary nodes have different temperatures, flow variations and reversals also affect the gas temperatures and the powers exchanged with the ambient at the prescribed temperature, as shown in figures 5 and 6, respectively.

Thanks to the adopted model structuring, the simulation of 30 minutes of operation for the modelled system took 60 milliseconds on a laptop with an i7-1165G7 CPU at 2.80GHz running Ubuntu Linux 22.04 LTS, using OpenModelica version 1.19.3 and clang version 14.0.0.

5.2 Example 2

The second example is more articulated, and involves control. The modelled system, depicted as Modelica diagram in Figure 7, is composed of a liquid cooling loop releasing heat to an external gas cooling system.

The liquid loop is composed first of a simple CPU model – (1) in Figure 7 – in the form of a lumped-parameter uniform 3D grid solid. Attached to the CPU are on one side an interface material layer (2) and on the other side a prescribed time-varying power (3), taken from a recording on a physical machine through (4). In a detailed simulation (1–4) would just need replacing with the component to

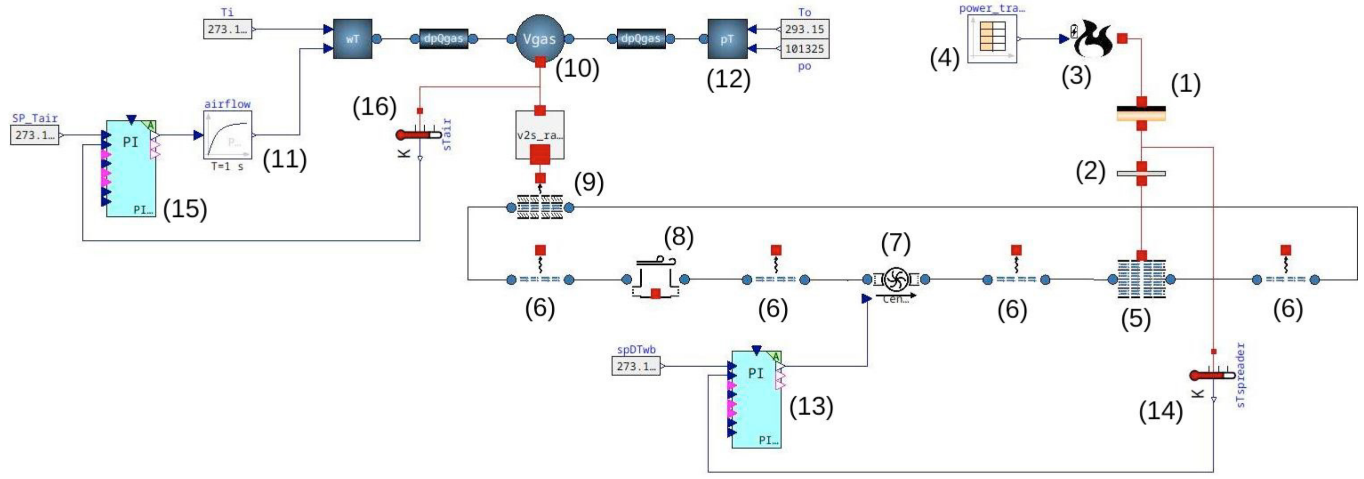

Fig. 7. Water/gas cooling loop with controls – Modelica diagram.

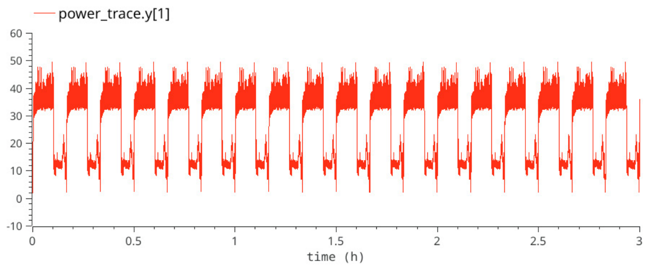

Fig. 8. Water/gas cooling loop with controls – applied power profile.

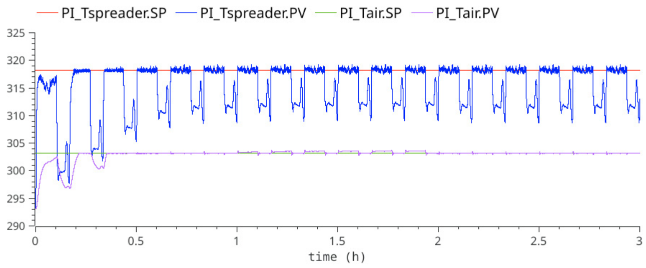

Fig. 9. Water/gas cooling loop with controls – temperatures and corresponding set points.

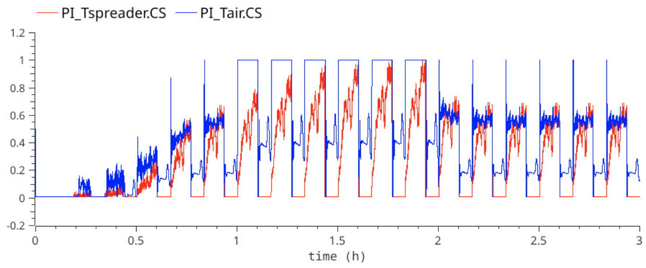

Fig. 10. Water/gas cooling loop with controls – control signals.

realise the interfacing with 3D-ICE or the external code of interest.

Coming back to the Modelica side, cooling is provided by the waterblock (5), in which water flows through a loop made of pipes (6), a centrifugal pump (7), a storage tank (8) and an exchanging pipe (9). This pipe releases heat to a gas volume (10) in which the air flowrate is commanded via a mechanism here simplistically represented by the first-order dynamics in (11). Air is discharged to the ambient at constant pressure (12).

Note that it would be possible to vary the air intake temperature and the discharge pressure, to simulate possible unforeseen stress for the cooling system. We do not report such tests for space limitation, but the matter is worth mentioning.

The control system is composed of two PI controllers.The former (13) acts on the pump to keep the temperature measured by sensor (14) near the spreader below a prescribed limit. The latter (15) manipulates the air flowrate command to (11) so as to keep the air outlet temperature measured by sensor (16) at the desired value.

Figures 9 and 10 report the result of a simulation test in which the power profile in Figure 8 was applied to the system. As can be seen, the system operates correctly: the set points are attained, and when the air flowrate command from the air temperature PI hits saturation, see the middle part of Figure 10, the spreader temperature PI naturally intervenes by increasing the water flowrate.

Concerning performance, the three hours of simulated time took 31 seconds on the same i7 laptop, which is approximately 350 times faster than real time and is an encouraging result if one considers that model has about 700 equations – several of which are nonlinear – and 84 continuous state variables.

6. CONCLUSIONS AND FUTURE WORK

This paper belongs to a long-term research about energy-efficient operation of computing systems, and in particular about innovative cooling solutions aimed to preserve the integrity and the reliability of the said systems while resulting in a minimal – or in any case acceptable – performance reduction.

While relating the presented work to the rest of our research and the wider literature *panorama*, we evidenced that the design and the assessment of modern cooling systems necessarily require simulation tools to address the entire process and the numerous decisions to take, both about choosing/sizing the equipment and about the controls that are inevitably required.

In this context, the contribution we presented is a Modelica library for computer cooling systems simulation. The library has a fully modular structure, is conceived for use by people who are experts more of the computing field than of simulation methods and tools, is open to inter-operation with external codes for both substance properties computation and detailed 3D thermal chip simulation, and was developed and tested with a free Modelica translator.

We also reported a couple of simulation examples to illustrate the capabilities – and, in particular, the computational efficiency – of the library, which will be soon released as free software for the scientific and engineering community.

Future work will be directed to extending the library, particularly with efficient models of thermal machines so as to address the higher levels of large-scale (e.g., data centre-wide) cooling solutions. We are also continuing the validation activities, and we hope that the library will foster collaborations on a matter that, in the future, will surely gain more importance than it already has.

REFERENCES

Aglawe, K., Yadav, R., and Thool, S. (2022). Current technologies on electronics cooling and scope for further improvement: A typical review. In *Proc. 2020 the International Conference on Industrial and Manufacturing Systems*, 389–408. Jalandhar, Punjab, India.

Ali, G., Wofford, L., Turner, C., and Chen, Y. (2022). Automating CPU dynamic thermal control for high performance computing. In *Proc. 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing*, 514–523. Taormina, Italy.

Bell, I.H., Wronski, J., Quoilin, S., and Lemort, V. (2014). Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp. *Industrial & Engineering Chemistry Research*, 53(6), 2498–2508.

Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Clauß, C., Elmqvist, H., Junghanns, A., Mauss, J., Monteiro, M., et al. (2011). The functional mockup interface for tool independent exchange of simulation models. In *Proc. 8th international Modelica conference*, 105–114. Dreden, germany.

Casella, F. and Leva, A. (2006). Modelling of thermohydraulic power generation processes using modelica. *Mathematical and Computer Modelling of Dynamical Systems*, 12(1), 19–33.

Casella, F. and Richter, C. (2008). ExternalMedia: a library for easy re-use of external fluid property code in Modelica. In *Proc. 6th International Modelica Conference*, 157–161. Bielefeld, germany.

Fritzson et al., P. (2020). The OpenModelica integrated environment for modeling, simulation, and model-based development. *Modeling, Identification and Control*, 41(4), 241–295.

Functional Mock-up Interface standard home page (2022). `https://fmi-standard.org/`.

Huang, W., Ghosh, S., Velusamy, S., Sankaranarayanan, K., Skadron, K., and Stan, M.R. (2006). Hotspot: A compact Thermal Modeling Methodology for Early-Stage VLSI Design. *IEEE Trans. Very Large Scale Integr. Syst.*, 14(5), 501–513. doi:10.1109/TVLSI.2006.876103.

Ladenheim, S., Chen, Y., Mihajlović, M., and Pavlidis, V.F. (2018). The MTA: An Advanced and Versatile Thermal Simulator for Integrated Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(12), 3123–3136.

Lemmon, E., Huber, M., and Mclinden, M. (2007). NIST standard reference database 23: reference fluid thermodynamic and transport properties – REFPROP, version 8.0.

Leva, A., Terraneo, F., Giacomello, I., and Fornaciari, W. (2017). Event-based power/performance-aware thermal management for high-density microprocessors. *IEEE Transactions on Control Systems Technology*, 26(2), 535–550.

Modelica Association home page (2022). `https://modelica.org/`.

OpenModelica Consortium home page (2022). `https://openmodelica.org/`.

P. Fritzson, P. (2014). *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*. John Wiley & Sons, New York, NY, USA.

Seuret, A., Iranfar, A., Zapater, M., Thome, J., and Atienza, D. (2018). Design of a two-phase gravity-driven micro-scale thermosyphon cooling system for high-performance computing data centers. In *Proc. 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 587–595. San Diego, CA, USA.

Sridhar, A., Vincenzi, A., Ruggiero, M., Brunschwiler, T., and Atienza, D. (2010). 3D-ICE: Fast compact transient thermal modeling for 3d ics with inter-tier liquid cooling. In *Prroc. 2010 IEEE/ACM International Conference on Computer-Aided Design*, 463–470. San Jose, CA, USA.

Terraneo, F., Leva, A., Fornaciari, W., Zapater, M., and Atienza, D. (2021). 3D-ICE 3.0: efficient nonlinear MPSoC thermal simulation with pluggable heat sink models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(4), 1062–1075.

Terraneo, F., Leva, A., and Fornaciari, W. (2019). Event-Based Thermal Control for High Power Density Microprocessors. In W. Fornaciari and D. Soudris (eds.), *Harnessing Performance Variability in Embedded and High-performance Many/Multi-core Platforms: A Cross-layer Approach*, 107–127.

Zhang, Q., Meng, Z., Hong, X., Zhan, Y., Liu, J., Dong, J., Bai, T., Niu, J., and Deen, M. (2021). A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization. *Journal of Systems Architecture*, 119, 102253.